

dossier

COMMODORE



**PROGRAMMI
TRUCCHI
DIDATTICA**

COMMODORE

dossier

Mensile
di giochi e programmi
per COMPUTER
COMMODORE

Direttore Responsabile
Marco Bindi

Direttore Tecnico:
Gianroberto Negri

Edizioni **CO.STA** srl
via Mazzini, 128
56100 PISA

Registrazione al Tribunale
di Pisa n. 2 del 22/2/84
Spediz. in abbon. post.
gruppo III/70

COMMODORE DOSSIER

il mensile per realizzare la tua Biblioteca
di Software in tre sezioni:

- Giochi
- Programmi personali
- Programmi di utilità

Lire 2000

I numeri arretrati, a lire 4000, possono
essere richiesti senza oneri di spedizione
con pagamento anticipato in contanti,
o tramite assegno o vaglia postale
intestato alla Società editrice.

Stampa:
Rotolito Lombarda Spa (MI)

Distribuzione
Messaggerie Periodici (MI)

**Nel numero
di maggio
di Personal Computer
una sorpresa:
PCMON 2
Un accattivante
monitor in Basic.
Non perderlo.**

**PER IL TUO
COMMODORE 64**



EASY COMPUTING

Via A. Bertani 24 - 50137 Firenze

EASY COMPUTING ha selezionato per te:

ABACUS

ZOOM PASCAL

È il primo compilatore PASCAL disponibile per CBM-64,
che crea veri codici macchina 6502.

SUPER DISK UTILITIES

Copia rapida in 4 versioni diverse: Total, Bam, Append,
Files.

SCREEN GRAPHICS

Estensione del BASIC COMMODORE 64. 24 comandi gra-
fici supplementari.

ULTRABASIC

È una versione più sofisticata di SCREEN GRAPHICS.

CHARTPAK

Consente di realizzare facilmente rappresentazioni gra-
fiche.

SYNTHY 64

Un interprete BASIC che permette di sfruttare al massi-
mo le capacità musicali del C-64.

TAS 64

Programma di gestione portafoglio.

CADPAK

Programma che permette di tracciare disegni e grafici con
l'aiuto della LIGHT PEN.

VIDEOBASIC, GRAPHICS DESIGNER, etc.

Richiedi il catalogo generale dei prodotti Easy Computing a:
EASY COMPUTING ITALIA, Via A. Bertani 24, 50137 Firenze.

Sono interessato a ricevere il catalogo generale EASY COMPU-
TING, gratuitamente e senza impegno, al seguente indirizzo:

Nome _____
Cognome _____
Indirizzo _____
Città _____ CAP _____
Professione _____
Tel. _____

di draghi, navi spaziali, alieni, mostri preistorici e ogni altra cosa che potrà venire in mente. Ritornando alla figura 2, avrete probabilmente notato che sulla destra d'essa sono presenti per ogni riga, tre numeri separati da una virgola; ebbene essi sono i valori da caricare in memoria mediante la parola chiave POKE. E' estremamente importante che essi vengano caricati in maniera sequenziale, riga dopo riga e non a caso, altrimenti la figura che ne scaturirà sarà assai diversa da quanto ci eravamo prefissi di realizzare. Come si arrivi ad essi e' presto detto ed e' molto piu' semplice di quanto possa sembrare a prima vista. Quando avevamo descritto la prima figura, se ben ricordate avevamo parlato dei numeri riportati sopra ad essa e che partendo da destra ed andando verso sinistra, raddoppiavano. Bene per ottenere i numeri riportati sulla destra, basterà sommare i numeri riportati sopra ad ogni quadratino annerito e questo per ogni riga. Ad esempio se nella prima riga in alto vi fossero tre quadratini anneriti e precisamente, partendo da destra il primo, il terzo ed il sesto; il numero da scrivere sulla destra sarebbe 37. Questo perché il numero riportato sopra al primo quadratino e' il numero 1, quello riportato sul terzo quadratino e' il numero 4 ed infine quello riportato sul sesto quadratino e' il numero 32. Ora se li sommiamo assieme otteniamo $1+4+32=37$ che e' appunto il numero da scrivere. Attenzione però, se osserviamo la figura, potremo notare che ogni otto quadratini, partendo da destra, vi e' una riga verticale piu' scura che attraversa l'intera scacchiera. Questo a causa del fatto che i numeri vanno sommati solo di otto quadratini in otto quadratini e non tutti quelli che si trovano sulla medesima riga. Altrimenti come da una unica riga potremmo ottenere 3 numeri?... Bene a questo punto abbiamo i valori da caricare in memoria, mediante la parola chiave POKE ma non sappiamo ancora DOVE debbono essere caricati. Rimane ovvio che vi siano delle zone di memoria atte a questo scopo ma quali sono?... E' presto detto: la zona di memoria utilizzata dal registratore per passare i dati da e al computer. Questa zona viene utilizzata solo in fase di salvataggio e di caricamento dei dati; poiché il programma che ci accingiamo a realizzare non utilizza questa gestione, se poniamo i nostri valori in questa zona di memoria, se ne staranno li' buoni, buoni sino a quando non utilizzeremo di nuovo il registratore. A questo punto non rimane altro da fare che tradurre in istruzioni e parole chiave quanto sino a questo punto e' stato detto ed analizzato.

```
1 RESTORE
5 M=832
10 FOR X= 0 TO 62 : READ N : POKE M+X,N : NEXT X
20 POKE 2040,13
1000 DATA 0,0,0,0,0,0,0,0
1010 DATA 0,0,0,0,0,0,0,0
```

```
1020 DATA 7,255,0,15,255,128
1030 DATA 15,255,192,31,255,224
1040 DATA 31,255,240,63,255,248
1050 DATA 127,255,255,127,255,0
1060 DATA 0,255,0,0,255,0,0,0
1070 DATA 0,0,0,0,0,0,0,0,0,0
```

Vediamo di analizzarlo dettagliatamente, istruzione dopo istruzione, parola chiave, dopo parola chiave. Iniziamo dall'istruzione 1 dove mediante la parola chiave RESTORE, ripristiniamo la lettura dei valori contenuti dopo le parole chiave DATA, alle istruzioni che iniziano dalla 1000 sino alla 1070. Proseguendo, troviamo l'istruzione 5 dove viene caricato il valore 832 nella variabile di tipo numerico M. Il valore numerico 832 che ora si trova nella variabile M e' l'indirizzo di memoria corrispondente alla zona riservata come e' stato prima detto, al registratore a cassette. Per quanti non si ricordino cosa sia un indirizzo di memoria basta che lo paragonino ad un reale indirizzo, quello ad esempio della casa nella quale abitano. Come l'indirizzo, della propria abitazione individua una e solo un'abitazione tra le numerose contenute in una città, così un indirizzo di memoria individua una ed una sola piccola zona di memoria, tra le numerose zone di memoria contenute in un computer. La prossima istruzione e' la 10, in essa troviamo le parole chiave FOR e TO che insieme alla variabile di tipo numerico X, generano un ciclo che va dal numero 0 sino al numero 62 per un totale di 63 volte. Di seguito mediante la parola chiave READ vengono via, via, letti i valori numerici presenti dopo la parola chiave DATA nelle istruzioni che iniziano dalla 1000 sino alla 1070, caricandoli nella variabile di tipo numerico N. Sempre nella medesima istruzione, mediante la parola chiave POKE i valori acquisiti mediante la parola chiave READ e la variabile di tipo numerico N, vengono caricati in memoria a partire dall'indirizzo 832 (M+X cioè 832 che e' il contenuto della variabile di tipo numerico M, viene sommato al valore della variabile di tipo numerico X, via, via incrementata dalle parole chiave FOR, TO e NEXT) sino all'indirizzo 895. La parola chiave NEXT e la variabile numerica X, chiudono il ciclo. Istruzione 20: in essa mediante la parola chiave POKE, viene caricato il valore 13 all'indirizzo di memoria 2040. E' estremamente importante soffermarci su questa istruzione e questo perché l'indirizzo di memoria 2040, e' quello che comunica al COMMODORE 64 la zona od indirizzo di memoria dal quale prelevare i dati da utilizzare nella creazione della SPRITE. Ora come si può facilmente osservare il valore caricato e' 13 e non come sembrerebbe logico 832, dato che quest'ultimo e' il reale indirizzo al quale abbiamo messo i valori da far utilizzare al COMMODORE 64 per la creazione della SPRITE. Infatti!... Ma non si tratta di un errore... Il valore 13 rappresenta sempre il valore 832.

(l'indirizzo al quale si trovano i valori della SPRITE), solo diviso per il numero 64. Se quindi moltiplichiamo il valore 13 per il valore 64, quale verifica di quanto appena detto, otteniamo appunto il valore 832. Questo artificio è voluto dal sistema operativo del COMMODORE 64. Attenzione!!... Nel caso utilizzassimo una zona di memoria il cui indirizzo è un numero assai più grosso e quindi il risultato della divisione d'esso per il numero 64, risultasse più grande di 255, non potremmo comunicare il valore al computer con un unico POKE alla locazione 2040 e questo a causa del fatto che in essa il numero più grande che potremmo caricare è appunto 255. Si renderà quindi necessaria un'altra tecnica di cui parleremo in un prossimo numero della rivista. Per questa volta, per non complicare troppo le cose, accontentiamoci d'utilizzare una unica SPRITE, i cui valori sono caricati a partire dall'indirizzo di memoria 832. Continuando nell'analisi del programma, le istruzioni che vanno dalla 1000 sino alla 1070, contengono, dopo la parola chiave DATA, i valori della nostra preziosissima SPRITE. Come già ribadito, essi vengono letti e caricati in memoria, mediante l'istruzione 10. A questo punto ci ritroviamo con i valori della SPRITE, caricati in memoria a partire dall'indirizzo 832 ed abbiamo comunicato al COMMODORE 64, mediante una POKE all'indirizzo 2040 ove essi si trovano. Il passo successivo sarà quello di mettere in condizione il computer, utilizzando quanto sino ad ora fatto per creare e muovere sullo schermo video la nostra SPRITE. Aggiungiamo perciò al nostro programma, le istruzioni che ci permetteranno di fare questo, come si può notare nel listato qui sotto riportato, ed analizziamole accuratamente per comprenderne in maniera semplice ma completa il corretto utilizzo.

```
1 RESTORE
5 M=832
10 FOR X= 0 TO 62 : READ N : POKE M+X,N : NEXT X
20 POKE 2040,13
```

```
-----
30 S=53248
40 POKE 53280,0:POKE53281,0
50 POKE S+21,1
60 POKE S+39,3
70 POKE S+1,50
80 FOR A= 1 TO 255 : POKE S+0,A: NEXT A
90 END
```

```
-----
1000 DATA 0,0,0,0,0,0,0,0,0,0
1010 DATA 0,0,0,0,0,0,0,0,0,0
1020 DATA 7,255,0,15,255,128
1030 DATA 15,255,192,31,255,224
1040 DATA 31,255,240,63,255,248
1050 DATA 127,255,255,127,255,0
1060 DATA 0,255,0,0,255,0,0,0,0
1070 DATA 0,0,0,0,0,0,0,0,0,0
```

Le istruzioni nuove, come si può facilmente notare dalle due righe che le separano da quelle precedenti sono quelle che vanno dalla 30 alla 90.

Istruzione 30: in essa viene caricato il valore 53248 nella variabile di tipo numerico S. Esso è l'indirizzo iniziale dei REGISTRI del COMMODORE 64 adibiti alla gestione delle SPRITE e della grafica. Il termine REGISTRO non spaventi, per il momento diciamo che è semplicemente un altro modo per chiamare una locazione di memoria, adibita a scopi un po' particolari. Per accedere ad ognuno dei 47 REGISTRI, basterà aggiungere il numero del REGISTRO desiderato all'indirizzo iniziale che come detto è 53248.

Istruzione 40: mediante la parola chiave POKE, vengono modificati i colori del fondo e del bordo del video, portandoli a nero.

Istruzione 50: in essa viene utilizzata la tecnica sopra descritta riguardo ad i REGISTRI del COMMODORE 64; infatti per individuare il REGISTRO numero 21, corrispondente all'indirizzo di memoria 53269, viene semplicemente aggiunto il numero 21 al contenuto della variabile di tipo numerico S. Ora poiché nella variabile di tipo numerico S è contenuto il valore numerico 53248, sommando 21 a 53248 si ottiene appunto 53269 che è l'indirizzo del REGISTRO a cui vogliamo accedere. Esso serve a comunicare al computer che vogliamo far apparire la SPRITE e quindi di renderla visibile.

L'istruzione 60, utilizzando la medesima tecnica, individua il REGISTRO numero 39, corrispondente all'indirizzo di memoria 53287. Esso serve per comunicare al computer il colore con cui desideriamo visualizzare la SPRITE ed in questo caso il colore CYAN.

L'istruzione 70, sempre facendo capo alla medesima tecnica, individua il registro numero 1 che serve per comunicare al computer la posizione VERTICALE alla quale desideriamo porre la nostra SPRITE ed in questo caso la posizione 50.

L'istruzione 80 è quella che finalmente ci permette di muovere la SPRITE in direzione ORIZZONTALE, partendo dalla posizione numero 1 sino ad arrivare alla posizione 255. Questo viene realizzato mediante un ciclo FOR-NEXT che provvede a far aumentare progressivamente il contenuto del REGISTRO numero 0 che ovviamente serve per comunicare al computer la posizione orizzontale alla quale vogliamo visualizzare la SPRITE.

L'istruzione 90 infine, mediante la parola chiave END fa fermare in maniera definitiva il programma. Il risultato che ci eravamo prefissati di raggiungere, quello cioè di creare e muovere una SPRITE in senso orizzontale sullo schermo video lo abbiamo infine raggiunto. Questa è una parte basilare per la realizzazione di giochi. L'apparente difficoltà d'utilizzazione delle SPRITE e' nostro augurio che sia stata almeno in parte risolta. Ovviamente di cose da

conoscere sulle SPRITE, ve ne sono ancora molte, come ad esempio il fatto che si possono utilizzare ben 8 SPRITE contemporaneamente a video e che ovviamente questo complica un po' le cose, rispetto a come le abbiamo viste ora. Sarà comunque compito di un

prossimo articolo dare accurate spiegazioni sul come utilizzare le altre tecniche offerte dalle SPRITE e questo sino alla realizzazione di un gioco completo che potrete orgogliosamente dire di aver realizzato con le Vostre mani... A presto.



Scopo di questa rubrica e' quello di prenderVi per mano e portarVi alla conoscenza dell'INFORMATICA, del linguaggio di programmazione BASIC e delle PAROLE CHIAVE che lo compongono. Il tutto realizzato finalmente in maniera semplice, chiara e molto dettagliata. Il fenomeno chiamato INFORMATICA DI MASSA che si sta' sviluppando attualmente in Italia ha portato ad un notevole proliferare di testi e riviste in cui si parla di linguaggio BASIC, di PAROLE CHIAVE, di PROGRAMMAZIONE... Pochi pero' trattano l'argomento in maniera veramente semplice e chiara. La caratteristica che si riscontra prevalentemente e' che sono troppo complicati per chi inizia a muovere i primi passi e di contro, troppo semplici per chi l'informatica e la programmazione gia' la conosce. E' normale trovare in essi, subito nelle prime pagine un programma in cui sono contenute molte delle parole chiave del BASIC, ed e' quindi normale che chi inizi ad avvicinarsi a quella meravigliosa scienza che e' l'INFORMATICA, possa sentirsi scoraggiato. Le cose vanno fatte per gradi, non si puo' "sparare" un programma e sperare che possa subito essere recepito da quanti leggono il libro o la rivista. E' appunto per aiutare questo stuolo di volenterosi che e' nata, come gia' detto questa rubrica dedicata al BASIC ed alle sue PAROLE CHIAVE. L'INFORMATICA e' anche una passione oltre che una scienza esatta: ed essendo appunto una passione e' aperta a tutti gli appassionati dai sei ai centoventi anni, ed a tutti i livelli immaginabili e possibili. Il suo unico limite e' costituito dalla fantasia di chi se ne occupa e poiche' il caso vuole che di fantasia ne possediamo tutti in grandissima quantita', lascio a Voi le conclusioni. Detto questo, non rimane altro da fare che iniziare a parlare dell'INFORMATICA e della PROGRAMMAZIONE che e' uno dei mattoni fondamentali della prima. PROGRAMMAZIONE... Puo' sembrare una parola difficile... Una di quelle parole che si collegano a cervelloni in canice bianco, seduti in immensi saloni, disseminati di COMPUTER dalle straordinarie propieta' e capacita'. Nulla di tutto questo, anzi Vi diro' di piu': forse non tutti sanno

che in effetti in ogni momento della nostra vita, stiamo in realta' PROGRAMMANDO!... A questo punto, saranno sorte molte domande in proposito, e per ovviare ad esse, sara' bene fare qualche esempio chiarificatore.

Immaginiamo quindi di aver deciso di andare a trovare un nostro parente; ed immaginiamo che egli abiti in una citta' a noi vicina. Ovviamente per poter arrivare da lui, dovremo necessariamente munirci di un mezzo di trasporto, salirvi sopra e recarci nella citta' dove abita. Giunti a destinazione dovremo cercare la via ed il numero dove e' situata la sua abitazione; fatto questo dovremo suonare alla porta ed attendere che ci venga ad aprire. Nel caso che, dopo un po' di tempo, non venga nessuno, dedurremo che non si trova in casa e ritorneremo indietro, altrimenti lo saluteremo ed entreremo in casa. Ebbene in realta' abbiamo PROGRAMMATO la visita al nostro parente; abbiamo cioe' elencato tutto un insieme di azioni da compiere per andare a fargli visita. Bene tra questo modo di PROGRAMMARE e quello che si compie su di un COMPUTER di qualsiasi forma e dimensione c'e' in realta' ben poca differenza. Anche su di un computer le ISTRUZIONI che gli vengono impartite, determinano un insieme di azioni che esso diligentemente esegue. Tutto nella nostra vita e' quindi in realta' PROGRAMMAZIONE. Lo e' ad esempio una ricetta di cucina, dove per poter realizzare il piatto che ci siamo prefissati, dobbiamo seguire tutto un insieme di precise ISTRUZIONI che ci dettagliano su cosa dobbiamo fare, affinche' il piatto riesca nella maniera voluta e non diventi un qualche cosa di immangiabile nel vero senso della parola. Anche nell'ambito della scuola vi e' un PROGRAMMA da seguire, inerente alle varie materie del corso di studi intrapreso. Quindi per far compiere ad un computer un qualcosa, bastera' dirgli passo dopo passo, una istruzione alla volta quello che desideriamo esso faccia. Rimane ovvio che il sistema utilizzato dagli esseri umani non gli e' comprensibile, percio' dovremo usare un linguaggio che possa essere da esso compreso e nel nostro caso e' il

BASIC.

Prima però d'iniziare ad analizzare le parole chiave di questo linguaggio è necessario fare una premessa ed introdurre il concetto di **VARIABILE**. Questo perché le **VARIABILI** sono estremamente importanti, anzi direi essenziali per poter programmare bene. Una volta chiarito cosa siano e a che cosa possano servire, il tutto si rivela molto più semplice da comprendere ed assimilare. Bene per poter spiegare cosa sia una **VARIABILE**, dovremo ricorrere alla nostra fervida fantasia e provare ad immaginare di avere davanti a noi, appoggiati sopra un tavolo, tanti piccoli contenitori con dentro qualcosa. A questo punto prendiamone uno, e poiché vogliamo distinguerlo dagli altri, diamogli un nome, magari scrivendolo sopra un'etichetta che poi attaccheremo sopra d'esso, ad esempio "**COMPONENTI ELETTRONICI**". Fatto questo, guardiamo dentro ed osserviamone il contenuto... Toh!... Contiene una **VALVOLA**!... E da dove è uscito un simile vecchiume?... Togliamola ed al suo posto mettiamo,

come giustamente deve stare un **TRANSISTOR**. Fatto ciò, rimettiamo al suo posto il contenitore. Quello che abbiamo appena fatto è assai importante, poiché il contenitore è sempre lo stesso, e così pure il nome "**COMPONENTI ELETTRONICI**" che gli abbiamo dato; mentre è **VARIATO** il suo contenuto. Infatti ora al posto di una valvola, si trova un transistor. Ovviamente al posto di un transistor, potevamo pure mettere qualsiasi altro componente elettronico; quello che è importante comprendere è il fatto che il contenuto di "**COMPONENTI ELETTRONICI**", può essere **VARIATO**, infinite volte, quindi il contenuto è **VARIABILE**!... A questo punto rimane evidente che il nome **VARIABILE** contraddistingue un **CONTENITORE** il cui contenuto può essere variato infinite volte, altrimenti non avrebbe senso chiamarlo **VARIABILE**. In un computer le **VARIABILI**, cioè i **CONTENITORI** ed il loro **CONTENUTO**, hanno una forma differente ma, per il momento non dobbiamo interessarcene, l'importante è comprendere che si comportano e possono essere utilizzate nella medesima maniera.



PRINT USING

Il **COMMODORE 16**, l'ultimo nato della prestigiosa casa americana e di cui questa rubrica si occupa, è un personal computer che, nonostante il basso prezzo, possiede potenza e capacità non riscontrabili in molti personal computer la cui fascia di prezzo è notevolmente più elevata. Ma essendo l'ultimo nato, poco si è ancora scritto su di esso, sulle sue caratteristiche e sulla sua programmazione. Ecco quindi il motivo di questa rubrica della rivista, interamente dedicata ad esso, dove troverete una analisi del suo **BASIC** ed ovviamente delle sue **PAROLE CHIAVE**, con programmi esemplificativi del loro uso. Vi troverete pure programmi per: il Vostro divertimento, le Vostre attività, per aiutarVi nell'uso del computer, per insegnare le varie materie ai Vostri fratelli o figli e tanti altri. A proposito: inviate, inviate ed ancora inviate i Vostri programmi per il **COMMODORE 16** ed il **PLUS/4**, i migliori saranno pubblicati. Una raccomandazione però: devono essere ben descritti e documentati, affinché, possano essere compresi da tutti. Potrete così far osservare orgogliosamente ai Vostri amici, la rivista

con il Vostro nome riportato sopra. Sono molti i possessori di questo computer, per cui ci aspettiamo che ci sommergete di programmi, trucchi, utilità ed altro. Detto ciò, passiamo all'argomento **BASIC** e **PAROLE CHIAVE** del **COMMODORE 16**. Non esamineremo le parole chiave basilari, poiché questo viene già fatto nell'apposita rubrica dedicata alla didattica dove queste vengono analizzate a fondo, ma soltanto quelle **PAROLE CHIAVE** non presenti nel **BASIC** del **COMMODORE 64** e del **VIC 20**. Prima di addentrarci nella loro analisi è interessante osservare una caratteristica veramente molto pratica, presente sul **COMMODORE 16**: il tasto di **HELP**. Esso si rivela di grande aiuto nello svolgimento dell'attività di programmazione e per quanti, essendo all'inizio, commettono ancora molti errori di **SINTASSI**, sbagliando a scrivere le parole chiave o dimenticando qualche altra cosa nelle varie istruzioni. Il suo funzionamento è presto detto: ipotizziamo di scrivere un programma come quello, di seguito riportato:

```
10 PRINT
20 PRINTN "PIPPQ"
30 PRINT
```

Come si può facilmente osservare, all'istruzione 30, al posto della parola chiave PRINT è stato scritto PRINN. Ovviamente, una volta che il programma sia stato lanciato, essendo PRINN una parola chiave errata, apparirebbe a video il messaggio: SINTAX ERROR; il quale tradotto in italiano significherebbe: ERRORE DI SINTASSI. Ora il fatto che abbiamo commesso un errore di sintassi è chiaro ma, a quale istruzione lo abbiamo commesso, e nell'ambito dell'istruzione, quale parola chiave abbiamo sbagliato?... A questo provvede il tasto HELP. Infatti al verificarsi di un errore di sintassi, basterà premerlo e come per magia, verrà visualizzata l'istruzione errata che lampeggia in continuazione!!!!... A questo punto, utilizzando i tasti di cursore, quelli con le frecce rivolte nelle quattro direzioni e situati in alto, prima del tasto INST DEL, potremo modificare la parola chiave e quale conferma dell'avvenuta correzione, premere il tasto contrassegnato dalla parola RETURN. A causa di questo fatto, il COMMODORE16 ed il suo quasi gemello PLUS/4 sono dei computer ideali da utilizzare a scopo didattico, in corsi di programmazione, nelle scuole ed in altre sedi ove si desidera insegnare l'INFORMATICA.

Fatta questa interessante premessa, passiamo senza indugiare oltre ad analizzare le potenti PAROLE CHIAVE, offerte dal BASIC 3.5 del COMMODORE 16. La prima d'esse che passerà sotto la nostra analisi è la PRINT USING. Così a prima vista può sembrare l'unione di due PAROLE CHIAVE, di cui una la PRINT conosciuta ed una la USING del tutto nuova. In effetti il comportamento di questa particolare PAROLA CHIAVE è molto simile al comportamento della PRINT, solo che offre veramente molte possibilità in più rispetto alla PRINT. Essa è stata introdotta dai progettisti del COMMODORE 16 per dare la possibilità di rendere i dati stampati sul video o sulla stampante molto più leggibili da parte di chi sta utilizzando il programma. E, questo per chi non se ne sia ancora accorto è di un'importanza veramente notevole.

Per dimostrare quanto sino ad ora detto, facciamo riferimento ad un programmino, dove metteremo a confronto i dati stampati mediante la parola chiave PRINT, con i dati stampati mediante la parola chiave PRINT USING, analizzandone le differenze. Ecco quindi il programma e di seguito la sua dettagliata analisi e spiegazione.

```
10 A$="COMMODORE-DOSSIER"
20 F$="*****"
30 PRINT A$
40 PRINT USING "*****";A$
50 PRINT USING F$;A$
```

COMMODORE-DOSSIER
COMMODORE

COMMODORE-DOSSIER

Iniziando dall'istruzione 10, in essa viene caricata la costante alfanumerica "COMMODORE-DOSSIER" nella variabile di tipo alfanumerico A\$. Continuando dalla istruzione 20, in essa viene caricata la costante alfanumerica "*****" nella variabile di tipo alfanumerico F\$. Molto probabilmente salta subito all'occhio che il numero di caratteri # utilizzato nell'istruzione 20, equivale al numero di caratteri utilizzato dalla costante "COMMODORE-DOSSIER" della istruzione 10. In effetti possono sembrare strani tutti quei caratteri # caricati nella variabile F\$ ed oltretutto nel medesimo numero della costante dell'istruzione 10. Niente paura, il carattere # è essenziale per utilizzare nel modo migliore la potenza della parola chiave USING, unita alla parola chiave PRINT. L'istruzione 30, messa per poter confrontare le differenze di stampa introdotte dall'utilizzo della parola chiave USING da quelle delle convenzionale PRINT, mediante la parola chiave PRINT, stampa a video il contenuto della variabile A\$ ed in questo caso viene evidenziata la costante "COMMODORE-DOSSIER", come si può anche notare dall'esempio di lancio riportato sotto al programma dopo i trattini. Nell'istruzione 40, mediante la parola chiave PRINT, unitamente alla parola chiave USING, viene stampato a video il CONTENUTO della variabile di tipo alfanumerico A\$. Il risultato lo si può osservare sempre nell'esempio di lancio del programma e... E' alquanto strano!... Ed il resto della costante "COMMODORE-DOSSIER", contenuta nella variabile di tipo alfanumerico A\$ dove è finito?... Qui se ne vede soltanto un pezzo, costituito dai 5 caratteri iniziali... Mediante l'istruzione 30, utilizzando la parola chiave PRINT, l'abbiamo stampata interamente, quindi vuol dire che essa è ancora contenuta in tale variabile, altrimenti anche l'istruzione 30 ne avrebbe stampato a video solo i suoi 5 caratteri iniziali. Quindi perché l'aver messo la parola chiave USING, ha prodotto un risultato così anormale e strano?... Niente paura, forti del fatto che ogni cosa ha la sua logica spiegazione, vediamo di fornirla anche per questo caso. Torniamo quindi ad analizzare l'istruzione 40, contenente la parola chiave USING e soffermiamoci sui caratteri # posti tra virgolette, di seguito alla parola chiave incriminata... Se li contiamo, scopriamo che... Sono esattamente cinque!!!!... Ora poiché, durante lo svolgimento del programma, il risultato dell'istruzione 40 è stato quello di stampare i primi cinque caratteri della costante alfanumerica "COMMODORE-DOSSIER", contenuta nella variabile di tipo alfanumerico A\$, appare ovvio che tra il numero dei caratteri # presenti nell'istruzione 40 ed il numero dei caratteri stampati a video, deve per forza esserci una connessione. Ed infatti c'è ed è la seguente: il numero di caratteri alfanumerici stampati a video è direttamente

proporzionale con il numero di caratteri # presenti dopo la parola chiave USING e posti tra apici. Quindi se volessimo stampare ad esempio i primi sette caratteri della costante "COMMODORE-DOSSIER", contenuta nella variabile di tipo alfanumerico A\$, dovremmo scrivere una istruzione fatta in questo modo:

```
40 PRINT USING "*****";A$
```

Ed il risultato sarebbe: COMMODO

Rimane quindi ovvio che nel caso volessimo stampare a video tutta la costante, dovremo mettere tanti caratteri # quanti sono quelli della costante e nel nostro caso 17 caratteri #. Onestamente sarebbe un po' scomodo ogni volta che scriviamo una istruzione in cui e' contenuta la parola chiave USING, scrivere tanti caratteri #, quanti sono i caratteri da stampare. Per ovviare a questo eccessivo onere di lavoro ecco quindi un interessante trucco che guarda caso e' stato anche utilizzato nel programma che stiamo analizzando in maniera approfondita. A questo scopo, osserviamo attentamente l'istruzione 50. Appare subito evidente una differenza con la precedente istruzione, utilizzando la parola chiave USING: mancano i caratteri # che servono a stabilire il numero dei caratteri da stampare ed al suo posto si trova una variabile di tipo alfanumerico e per l'esattezza la variabile F\$. Se ben ci ricordiamo essa e' la variabile, contenuta nell'istruzione 20 ed in cui abbiamo caricato la costante "*****", costituita da 17 caratteri # che guarda caso corrispondono all'esatto numero dei caratteri della costante alfanumerica "COMMODORE-DOSSIER" che desideriamo stampare interamente a video. Il trucco e'

appunto questo: caricare una ed una sola volta tanti caratteri #, sotto forma di costante alfanumerica, quanti sono i caratteri massimi della costante o variabile che desideriamo stampare in una variabile di tipo alfanumerico, ad esempio F\$ come nel nostro esempio, ed utilizziamo tale variabile al posto dei caratteri #. Il risultato, come si puo' osservare facilmente dall'esempio riportato e' esattamente lo stesso, abbiamo cioe' ottenuto di stampare l'intera costante di tipo alfanumerico "COMMODORE-DOSSIER". Molti a questo punto si chiederanno la reale utilita di questa parola chiave, visto che per ottenere il medesimo risultato della notissima PRINT, abbiamo dovuto faticare un po'... E' presto detto!... Molte volte nelle applicazioni di tipo gestionale il contenuto di una variabile deve essere visualizzato in una precisa posizione del video e con un numero di caratteri definito in precedenza, ad esempio 10; questo a causa del fatto che di lato sono presenti altre scritte che se stampassimo una variabile con piu' di 10 caratteri, essa si sovrapporrebbe alle scritte. Come quindi limitare la lunghezza dei vari contenuti che una variabile puo' avere durante lo svolgimento di un programma se non ricorrendo alla parola chiave USING, che guarda caso e' stata creata proprio con questo scopo?... La reale potenza offerta da essa, non e' quindi completamente percepibile, ricorrendo ai brevi esempi riportati in questo articolo. Sara' comunque compito dei prossimi articoli, farla comprendere in maniera piu' approfondita, ricorrendo anche al suo utilizzo con le variabili di tipo numerico in cui essa mostra tutta la sua reale versatilita' e potenza. A rileggerci quindi nel prossimo numero dove continueremo nella sua dettagliata analisi.



Inizia da questo numero una rubrica dedicata ai trucchi ai piccoli programmi, alle cose nuove che vengono trovate o scoperte nel quotidiano utilizzo dei computer della COMMODORE. Saranno poche righe per ogni argomento, ma questo non significa che avranno poco valore, anzi potranno renderVi l'utilizzo del computer, molto piu' semplice e proficuo. Le prime righe di trucchi, programmi, piccole utilita' e scoperte varie le metteremo noi... Le prossime e ci auguriamo che siano veramente numerose, le metterete Voi e saranno il frutto delle tante ore passate sopra alla tastiera del Vostro affezionato computer COMMODORE.

*A1 CANCELLAZIONE AUTOMATICA D'ISTRUZIONI

Una delle istruzioni che mancano sul COMMODORE 64 e sul VIC 20 e' l'istruzione DELETE. Mediante essa e' possibile cancellare istruzioni con grande facilità, e questa caratteristica si rivela assai preziosa nella stesura di programmi di una certa complessità. Per ovviare a tale mancanza ecco un micro programma composto da una sola istruzione. Va digitato nella esatta maniera di come e' qui sotto riportato, in caso contrario non funzionerebbe.

```
1 ?(SHIFT CLR)[3 SU]*F=?*F="F+1":L="L":IF F<=L THEN[1
HOME]:POKE198,2:POKE631,13:POKE632,13:END
```

Attenzione perche' una volta digitata, questa istruzione sara' di difficile modifica a causa del fatto che il numero massimo di caratteri accettati dal COMMODORE 64 e' 80. Ora poiche' i punti interrogativi rappresentano la forma abbreviata della parola chiave PRINT, nel caso vogliate rivedere l'istruzione appena digitata, Vi accorgete che in essa al posto dei punti sono ora presenti le parole chiave PRINT e che quindi essa supera i fatidici 80 caratteri. L'istruzione appena digitata, andra' posta in cima ai Vostri programmi, caricandola dalla cassetta o dal dischetto, prima d'iniziare a scriverli e facendo attenzione a non scrivere istruzioni che portino il medesimo numero d'essa: il numero 1. Quando si presentera' la necessita' di cancellare delle istruzioni dal programma, bastera' dare il seguente comando diretto:

```
F=30:L=70:GOTO 1
```

E verranno cosi' cancellate l'istruzioni che vanno dalla 30 alla 70. Ovviamente per cancellare altre istruzioni, bastera' porre nella variabile F il numero d'istruzione dalla quale iniziare a cancellare e nella variabile L il numero d'istruzione compresa sino alla quale cancellare. Funziona sia sul COMMODORE 64 che sul VIC 20.

*A2 ORDINAMENTO

Questa piccola subroutine Vi permettera' di ordinare delle matrici monodimensionali, composte cioe' da tante variabili, individuabili ognuna mediante un numero progressivo che funge da indice. Essa potra' essere inclusa nei Vostri programmi e richiamata, mediante un GOSUB 40000. I dati da ordinare andranno preventivamente caricati nella matrice S0\$() ed il valore massimo che assumera' l'indice andra' caricato nella variabile di tipo numerico S1. Questo prima di richiamare la subroutine.

```
40000 S2=S1
40001 S2=INT(S2/2): IF S2=0 THEN 40009
40002 S3=1: S4=S1-S2
40003 S5=S3
40004 S6=S5+S2: IF S0$(S5)<S0$(S6) THEN 40007
40005 S7=S0$(S5):S0$(S5)=S0$(S6):S0$(S6)=S7:S5=S5-S2:
IF S5<1 THEN 40007
40006 GOTO 40004
40007 S3=S3+1: IF S3>S4 THEN 40001
40008 GOTO 40003
40009 RETURN
```

Di seguito e' riportato un programma esemplificativo che Vi permettera' di testare il funzionamento della subroutine e del quale daremo dettagliata spiegazione.

```
5 DIM S0$(9)
10 FOR I=0 TO 9:READ S0$(I):PRINT I;S0$(I):NEXT I:PRINT
20 S1=9:GOSUB 40000:FOR I=0 TO 9:PRINT I;S0$(I):NEXT I
:END
30 DATA ZETA,H,A,L,F,Q,P,T,B,D
```

Istruzione 5: in essa mediante la parola chiave DIM, viene dimensionata una variabile ad indice a 9. Attenzione: questo dimensionamento va' modificato se nel vostro programma i dati da ordinare sono maggiori di 9 ad esempio 200. L'istruzione 5 diverra' cosi':
5 DIM S0\$(200).

Istruzione 10: in essa viene generato un ciclo che inizia dal valore 0 sino al valore 9, per un totale di 10 volte; internamente al ciclo, mediante la parola chiave READ, vengono caricati i valori contenuti nell'istruzione 30 dopo la parola chiave DATA nelle variabili ad indice S0\$(), via, via incrementate dal ciclo, e contemporaneamente vengono anche stampati a video.

Istruzione 20: viene caricato nella variabile di tipo numerico il valore 9, corrispondente al massimo valore che potra' assumere l'indice; viene a questo punto, mediante la parola chiave GOSUB, richiamata la subroutine che si occupa di ordinare i valori che preventivamente sono stati caricati nelle variabili ad indice S0\$() mediante l'istruzione 10. Una volta che la routine ha provveduto ad ordinarli, mediante le parole chiave FOR, TO, PRINT e NEXT viene generato un ciclo che stampa i valori contenuti nella variabile ad indice S0\$() ormai ordinati a video.

Istruzione 30: in essa dopo la parola chiave DATA, sono contenuti i valori utilizzati nell'esempio riportato sopra. Rimane ovvio che i valori e/o dati da ordinare possono essere prelevati da FILES su cassetta o dischetto. L'importante e' che essi vengano caricati nella variabile ad indice S0\$() e che il massimo valore che potra' assumere l'indice, venga caricato nella variabile di tipo numerico S1.

*A3 SIMPATICO SUONO

Un programma di una linea per produrre il suono argenteo di un campanello che potrete inserire dove meglio crederete. Inizialmente viene caricato il valore 54272 nella variabile di tipo numerico S. Esso corrisponde all'indirizzo di memoria al quale iniziano i REGISTRI del COMMODORE 64 adibiti alla gestione del suono. Viene poi aggiunto il valore 24 al contenuto della variabile di tipo numerico S, ottenendo il valore 54296, rappresentante il REGISTRO adibito al controllo del volume e mediante la parola chiave POKE, viene

caricato in esso il valore 15 ottenendo così il massimo volume ottenibile. A questo punto viene caricato il valore 110 nel REGISTRO adibito al controllo della frequenza della nota (REGISTRO 54273) ottenendo la nota LA. Viene caricato il valore 9 nel registro adibito al controllo dell'Attack e del Decay (REGISTRO 54277). Viene caricato il valore 9 nel registro adibito al controllo del Sustain e del Release (REGISTRO 54278). Viene caricato prima il valore 17 e poi il valore 16 nel registro adibito al controllo della forma d'onda (REGISTRO 54276). Il risultato di tutto ciò è appunto il suono di un campanello.

10 S=54272:POKE S+24,15:POKE S+1,110:POKE S+5,9:POKE S+6,9:POKE S+4,17:POKE S+4,16

*A4 PULSANTE DI FUOCO

Nella creazione di giochi e programmi si rende necessario alcune volte di far fermare l'esecuzione del programma, sino al momento che non venga premuto ad esempio un tasto. Normalmente per realizzare un simile controllo si rendono necessarie diverse istruzioni. Non tutti però sanno che esiste un modo più sbrigativo per realizzare un simile controllo, utilizzando oltretutto il pulsante di fuoco del JOYSTICK. Esso utilizza una parola chiave del BASIC del COMMODORE 64 che in verità è poco conosciuta e poco utilizzata: la parola chiave WAIT. Per far fermare il computer sino al momento che verrà premuto il pulsante di fuoco del JOYSTICK basterà scrivere la seguente istruzione:

20 WAIT 56464,16,16

Questo, nel caso che il joystick sia connesso alla porta numero 2; nel caso della porta numero 1, la istruzione 20 diverrà:

20 WAIT 56465,16,16

Ecco un programmino che permetterà d'esemplificare maggiormente il tutto dando per scontato che il joystick sia connesso alla porta numero 2.

```
5 PRINT CHR$(147);
10 PRINT "PER CONTINUARE PREMERE IL PULSANTE DI FUOCO"
20 WAIT 56464,16,16
30 PRINT "OTTIMO!... BEN FATTO!..."
40 FOR A=1 TO 1000: NEXT A
50 GOTO 5
```

Eccone la spiegazione: Istruzione 5, in essa mediante la parola chiave PRINT ed il carattere il cui codice è 147, viene cancellato quanto in quel momento è presente sul video. Istruzione 10, mediante la parola chiave PRINT viene stampato a video il messaggio "PER CONTINUARE PREMERE IL PULSANTE DI FUOCO".

Istruzione 20, in essa mediante la parola chiave WAIT, viene fatto fermare il computer, sino a quando non viene premuto il pulsante di fuoco del joystick connesso alla porta numero 2. Istruzione 30, mediante la parola chiave PRINT, viene stampato a video il messaggio "OTTIMO!... BEN FATTO!...". Istruzione 40, mediante le parole chiave FOR, TO e NEXT, viene generato un ciclo da 1 a 1000 per dare il tempo di leggere il messaggio precedentemente stampato a video. Istruzione 50, mediante la parola chiave GOTO, viene effettuato un salto all'istruzione 5, alla quale ricomincia il programma appena esaminato.

*A5 LAMPEGGIO TOTALE

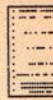
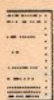
Si presenta talvolta la necessità di evidenziare fortemente un messaggio stampato sul video. A questo scopo è stato creato questo programma di una linea di cui la prima versione è per il VIC 20, la seconda è per il COMMODORE 64.

1 A=36879:POKEA,B:FORT=1 TO 99:NEXTT: POKE A,25:GOTO 1

1 A=53280:B=A+1:POKEA,0:POKEB,0:FORT=1 TO 99:NEXT T:POKE A,1:POKE B,1:GOTO 1

*A6 ANTI RESET

Nel COMMODORE 16 e nel PLUS 4 esiste un tasto di RESET che permette di ripristinare la condizione iniziale che si ha quando il computer viene acceso. Ovviamente il programma che in quel momento si trovava in memoria viene perso; ma questo non sempre fa piacere. Per avviare ad un simile inconveniente ecco un semplicissimo trucco: per prima cosa contemporaneamente al tasto RESET premere il tasto RUN/STOP, questo Vi porterà nel MONITOR, la parte del sistema operativo che Vi permette di scrivere programmi in linguaggio macchina. A questo punto battete la lettera X ed il tasto RETURN. Vi troverete di nuovo nel BASIC. Ora battete LIST ed il tasto RETURN... Il programma BASIC, non è stato cancellato!!!!.



PROGRAMMI

**VC
20**

SUONI

**UU
UU**

Una biblioteca d'effetti pronta per essere utilizzata nell'ambito dei vostri programmi. Al lancio del programma verra' visualizzato un menu', mediante il quale potrete scegliere l'effetto di cui avete bisogno. Dell'effetto scelto, verranno anche evidenziate a video le istruzioni che lo hanno prodotto. Questo per dare la possibilita' a chi utilizza le varie routine di sapere quale parte del programma sta' generando il particolare effetto che in quel momento sta' ascoltando. Il programma funziona sulla versione base del VIC20. Attenzione!!... Se nel listato trovate questo particolare carattere ^, esso equivale al segno dello elevamento a potenza e che sui vari commodore si ottiene mediante la freccia rivolta verso l'alto.

SPIEGAZIONE PROGRAMMA

Istruzioni 10-40: Viene creato a video il menu' delle varie opzioni previste dal programma.

Istruzioni 50-60: Viene prelevato un carattere dalla tastiera, corrispondente all'opzione scelta da chi sta utilizzando il programma. Il carattere prelevato mediante l'istruzione 50, viene mediante la parola chiave VAL, convertito in valore numerico per poter essere utilizzato per far saltare il computer alle varie parti del programma, adibite alla realizzazione dei suoni.

Istruzioni 70-90: Mediante queste istruzioni dopo che l'opzione scelta e' stata eseguita, viene fatto saltare il computer alla parte di programma che si occupa di gestire il menu'.

Istruzioni 100-740: Vengono gestite le varie routine che si occupano ognuna della generazione dei vari effetti.

Istruzioni 2000-2015: Viene visualizzata sul video la parte di programma che si occupa della generazione del particolare effetto richiamato.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : SUONI

VAR. : NUMERO ISTRUZIONE

A\$	50	60	80			
J	720					
L1	100	200	300	400	500	600
	700	2000				
L2	100	200	300	400	500	600
	700	2000				
N	120	220	240	320	420	620
	720					
S	220	320	420	620		
S1	230					
V	510					

```

10 REM
20 PRINT CHR$(147)" LIBRERIA SUONI",,,
"[2 SPZ]1 MACCHINA[2 SPZ]"
30 PRINT"[2 SPZ]2 LASER[7 SPZ]":PRINT
"[2 SPZ]3 SILURO[9 SPZ]":PRINT"[2 SPZ]4
UFO CHE ATTERRA"
40 PRINT"[2 SPZ]5 VECCHIO AERO[5 SPZ]"
,"[2 SPZ]6 BOMBA[2 SPZ]","[2 SPZ]7 SIR
ENA",,,,"[4 SPZ]SCEGLI[2 SPZ]"
50 GET A$:IF A$=""THEN 50
60 ON VAL(A$)GOTO 100,200,300,400,500,
600,700
70 POKE 214,15:PRINT" PREMI UN TASTO"
80 GET A$:IF A$=""GOTO 80
90 GOTO 20
100 L1=110:L2=130:GOSUB 2000
110 POKE 36878,15
120 FOR N=135 TO 200 STEP.5:POKE 36875
,N:POKE 36876,N
130 POKE 36875,0:POKE 36876,0:NEXT
140 GOTO 70
200 L1=210:L2=240:GOSUB 2000
210 POKE 36878,15
220 FOR N=1 TO 6:S=241:POKE 36876,S:PO
KE 36876,0:S=S-1:POKE 36876,S
230 S1=135:POKE 36875,S1:POKE 36875,0:
S1=S1+1:POKE 36875,S1
240 POKE 36875,0:POKE 36876,0:NEXT N
250 GOTO 70
300 L1=310:L2=320:GOSUB 2000
310 POKE 36878,15

```

```

320 FOR N=1 TO 3:FOR S=241 TO 135 STEP
-3:POKE 36875,S:POKE 36875,0:NEXTS:NEX
TN
330 GOTO 70
400 L1=410:L2=430:GOSUB 2000
410 POKE 36878,15
420 FOR N=1 TO 5:FOR S=200 TO 220+N*2:
POKE 36876,S:NEXT S:NEXT N
430 POKE 36876,0:POKE 36878,0
440 GOTO 70
500 L1=510:L2=520:GOSUB 2000
510 FOR V=13 TO 0 STEP-.02:POKE 36878,
V
520 POKE 36877,212:POKE 36877,0:NEXT
530 GOTO 70
600 L1=610:L2=630:GOSUB 2000

```

```

610 POKE 36878,15
620 FOR S=230 TO 128 STEP-1:POKE 36876
,S:FOR N=1 TO 20:NEXT N:NEXT S
640 GOTO 70
700 L1=710:L2=730:GOSUB 2000
710 POKE 36878,15
720 FOR N=1 TO 5:POKE 36875,200:FOR J=
1 TO 500:NEXT J
740 GOTO 70
2000 PRINT CHR$(147)CHR$(5)"RETURN":PR
INT"LIST";L1;"-";L2;
2010 POKE 631,19:POKE 632,17:POKE 633,
31:POKE 634,13
2015 POKE 635,19:POKE 636,13:POKE 198,
6:END

```

**VC
20**

TONI

**PP
PP**

Un ausilio musicale per aiutare i bambini a distinguere i vari toni musicali. Sono state previste tutte le possibili combinazioni dei TONI ALTI, MEDI e BASSI. Al lancio del programma verra' proposto al bambino di immettere una possibile combinazione, ad esempio AMB che significa ALTO, MEDIO e BASSO. Il programma provvedera' a questo punto a far ascoltare la combinazione prescelta.

DESCRIZIONE PROGRAMMA

Istruzioni 10-117: Mediante queste istruzioni viene gestita la parte di programma che si occupa del menu' e delle varie scelte, visualizzando anche alcune eventuali combinazioni tra quelle disponibili.

Istruzione 120: Viene accettato un dato in ingresso, ed in questo caso la combinazione di tre lettere che nelle istruzioni seguenti, determineranno la combinazione dei tre toni: ALTO, BASSO e MEDIO.

Istruzioni 125-150: Vengono testate le varie combinazioni di toni. Sofferamoci ad esempio sull'istruzione 125, in essa vi sono tre variabili: S1, S2 ed S3. Servono a stabilire la sequenza dei toni: S1 gestisce il primo tono, S2 il secondo ed S3 il terzo. Facendo riferimento sempre a questa istruzione, essendo la combinazione AMB che sta' per ALTO, MEDIO e BASSO nella variabile S1 verra' caricato il valore H cioe' 36876, nella variabile S2 verra' caricato il valore M cioe' 36875 ed infine nella variabile S3 verra' caricato il valore L e cioe' 36874. Il valore H significa ALTO, il valore M significa MEDIO ed il valore L significa BASSO.

Istruzioni 153-155: Viene gestita la continuazione o meno del programma.

Istruzioni 200-275: Viene gestita la parte vera

e propria del programma che si occupa della gestione dei toni musicali ALTO, BASSO e MEDIO. In essa a fronte delle combinazioni testate nelle istruzioni che vanno dalla 125 alla 150, vengono generati i toni, mediante il generatore musicale a tre voci del VIC20, leggendo i valori corrispondenti alle note dai DATA delle istruzioni che vanno dalla 400 alla 470 e gestendo appunto il generatore con essi.

Istruzioni 400-470: Sono qui' contenuti i valori letti e gestiti dalla parte di programma che va dall'istruzione 125 all'istruzione 150.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : TONI

VAR. | NUMERO ISTRUZIONE

VAR.	NUMERO ISTRUZIONE
C	40 125 130 135 140 145
	150 153
D	260 275
H	112 125 130 135 140 145
	150
L	112 125 130 135 140 145
	150
M	112 125 130 135 140 145
	150
N	275
P	220 230 250 270
Q	220 235 270
R	220 240 270
S1	125 130 135 140 145 150
	250 270
S2	125 130 135 140 145 150
	250 270

```

S3 : 125 130 135 140 145 150
    : 250 270
T : 153
V : 112 210
X : 205 270
X$ : 120 125 130 135 140 145
    : 150 153 154 205
Y : 205 230 235 240

```

```

10 REM
20 PRINT"[CLR][PORP][RVS ON]TONI,TONI
MUSICALI. IMPARIAMO A CONOSCERLI[2 SPZ
]"
30 PRINT"[BNC][RVS ON]SEGLIAMO LA COMB
INAZIONE TRA QUELLE DISPONIB."
40 C=36879:POKEC,27
100 REM
110 PRINT"[ROSS][RVS ON]SEGLIAMO... SC
EGLIAMO... SCEGLIAMO .....[2 SPZ]"
112 H=36876:M=36875:L=36874:V=36878
113 PRINT"[NERO]*****
*****[GIU']"
114 PRINT"[RVS ON][ROSS]AMB[RVS OFF] =
ALTO MEDIO BASS.[RVS ON]ABM[RVS OFF]
= ALTO BASS.MEDIO"
115 PRINT"[RVS ON]BMA[RVS OFF] = BASS.
MEDIO AL."
116 PRINT"[RVS ON]BAM[RVS OFF] = BASS.
ALTO MEDIO [RVS ON]MAB[RVS OFF] = MEDI
O ALTO BAS."
117 PRINT"[RVS ON]MBA[RVS OFF] = MEDIO
BASS.ALTO":PRINT"[RVS ON]FINE[RVS OFF
] = FINE PROGRAMMA[PORP][GIU']"
120 INPUTX$
125 IFX$="AMB"THENS1=H:S2=M:S3=L:POKEC
,60:GOTO200
130 IFX$="ABM"THENS1=H:S2=L:S3=M:POKEC
,234:GOTO200
135 IFX$="BMA"THENS1=L:S2=M:S3=H:POKEC
,90:GOTO200
140 IFX$="BAM"THENS1=L:S2=H:S3=M:POKEC
,138:GOTO200
145 IFX$="MAB"THENS1=M:S2=H:S3=L:POKEC
,156:GOTO200
150 IFX$="MBA"THENS1=M:S2=L:S3=H:POKEC
,174:GOTO200
153 IFX$="FINE"THENFORT=1TO90:PRINT"[G
IU'] [4 SPZ][ROSS]CIAO AMICO!!";POKEC,

```

```

T:NEXT
154 IFX$="FINE"THENPRINT"[CLR]":END
155 GOTO20
200 REM
205 PRINT"[NERO][RVS ON] "X$;" NUOVO A
SCOLTO!":X=200:Y=-100
210 POKE V,10
220 READ P,Q,R
230 IFP=0THENP=Y
235 IFQ=0THENQ=Y
240 IFR=0THENR=Y
250 IFP=-1THENPOKES1,0:POKES2,0:POKES3
,0:RUN
260 READ D
270 POKES1,P+X:POKES2,Q+X:POKES3,R+X
275 FORN=1TO9:NEXTN:GOTO220
400 REM
410 DATA 25,0,0,29,0,0,0,1,25,0,0,29,0
,0,0,1
415 DATA 25,0,0,20,28,0,0,10,31,0,0,29
,0,0,1
420 DATA 31,25,0,20,28,25,0,9,28,0,0,1
,31,25,0,20,32,25,0,10
425 DATA 32,0,0,1,35,25,0,20,35,28,0,1
0,35,31,0,29,0,0,1
430 DATA 40,31,25,10,0,31,25,10,40,28,
25,9,40,28,0,1
435 DATA 35,31,25,10,0,31,25,10,35,32,
25,9,35,32,0,0
440 DATA 31,35,25,10,0,35,25,10,31,35,
28,10
445 DATA 25,35,31,10,0,35,31,10,25,35,
31,9,25,35,0,1
450 DATA 35,40,31,10,35,0,31,10,32,40,
28,10,31
453 DATA 35,31,10,31,0,31,10,28,35,32,
10
455 DATA 25,31,35,10,25,0,35,10,25,31,
35,10,25
457 DATA 25,35,10,25,0,35,10,25,25,35,
10
460 DATA 0,35,40,10,0,35,0,10,0,32,40,
10,0,31
462 DATA 35,10,0,31,0,10,0,28,35,10
465 DATA 0,25,31,10,0,25,0,10,0,25,31,
10,0,25,25,10,0,25,0,10,0,25,25,10
470 DATA 0,0,35,20,0,0,32,10,0,0,31,20
,0,0,28,10,0,0,25,60,-1,-1,-1

```

**CM
64**

CONVERT

**UU
UU**

Per convertire i caratteri grafici e di controllo del
COMMODORE 64 in sigle stampabili mediante qualsiasi

stampante, bastera' digitare questo programma e
lanciarlo. Fino a quando non spegnerete il computer o
premerete i tasti RUN/STOP e RESTORE, qualsiasi
programma che caricherete in memoria, verra'
convertito. Bastera' a questo punto dare i seguenti
comandi in modo diretto: OPEN3,4:CMD3:LIST ed il

programma verrebbe stampato con i caratteri grafici e di controllo convertiti in sigle.

DESCRIZIONE PROGRAMMA

ISTRUZIONE 10: Viene visualizzato il messaggio "CONV.LISTATI" ed il messaggio "CARICAMEN.DATA".

ISTRUZIONI 20-40: Viene letto dalle locazioni di memoria 55 e 56 il punto in cui inizia il BASIC, abbassato il punto iniziale del basic di 630 BYTES per poter far posto alla parte di programma realizzata in linguaggio macchina che si occupa di convertire i caratteri grafici e di controllo nelle sigle che gli abbiamo fornito nelle istruzioni che vanno dalla 1010 alla 1350.

ISTRUZIONI 50-90: Viene caricata in memoria la prima parte di linguaggio macchina e viene anche controllato che i valori da caricare in memoria e presenti dopo la parola chiave DATA, nelle istruzioni che vanno dalla 5000 in poi siano esatti. In caso di valore errato, viene visualizzata a video il numero dell'istruzione in cui è contenuto il valore errato.

ISTRUZIONI 100-160: Mediante queste istruzioni, vengono caricati i valori in memoria, prelevandoli dalle restanti istruzioni DATA, unitamente al codice del carattere ed alla descrizione da noi fornita nelle istruzioni che vanno dalla 1010 alla 1350.

ISTRUZIONE 1000: Viene aumentato di 373 il valore della variabile B, corrispondente alla locazione in cui caricare i valori prelevati dalle istruzioni DATA nella seconda fase di caricamento ed il tutto viene caricato nella variabile di tipo numerico A.

ISTRUZIONI 1010-1350: Sono queste le istruzioni in cui sono contenute le sigle che sostituiranno i corrispondenti caratteri grafici e di controllo.

Nella prima parte di ognuna di queste istruzioni è caricato nella variabile di tipo numerico C il codice ASCII del carattere; nella seconda parte è caricato nella variabile di tipo alfanumerico C\$ la sigla che andrà a sostituire il carattere in memoria; infine nella terza ed ultima parte viene richiamata la routine che si trova a partire dall'istruzione 4000 fino alla 4010. E' mediante questa routine che le sigle vengono realmente caricate in memoria, sostituendole ai vari caratteri grafici e di controllo.

ISTRUZIONE 1998: Viene controllato che non siano stati utilizzati troppi caratteri per le sigle. Questo controllo si rende necessario per non rischiare di terminare la memoria predisposta a questo scopo.

ISTRUZIONI 4000-4010: Come già detto prima, esse si occupano di sostituire in memoria le sigle ai caratteri grafici e di controllo.

ISTRUZIONI 5000-5640: Sono contenuti in queste istruzioni, dopo le parole chiave DATA, i valori da caricare in memoria nelle varie fasi di caricamento previste dal programma.

E' importante ricordare che nel caso vengano premuti i tasti di RUN/STOP e RESTORE il programma di conversione andrà ricaricato per poter di nuovo funzionare.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : CONVERTITORE

VAR. | NUMERO ISTRUZIONE

A	40	70	1000	1998	4000	4010
B	20	30	40	120	140	1000
	1998					
B1	30	120				
B2	30	120				
C	1010	1020	1030	1040	1050	1060
	1070	1080	1090	1095	1100	1110
	1120	1130	1140	1150	1160	1170
	1180	1190	1200	1210	1220	1230
	1240	1250	1260	1270	1280	1290
	1295	1300	1310	1320	1325	1330
	1340	1350	4000			
C\$	1010	1020	1030	1040	1050	1060
	1070	1080	1090	1095	1100	1110
	1120	1130	1140	1150	1160	1170
	1180	1190	1200	1210	1220	1230
	1240	1250	1260	1270	1280	1290
	1295	1300	1310	1320	1325	1330
	1340	1350	4000			
I	60	70				
L	50	80	90	100	120	
S	50	70				
V	60	70	80	110		
X	110	120				
Y	110	120				
Z	4000					

```

10 PRINT"[CLR]CONV.LISTATI ":PRINT"CARICAMEN.DATA":PRINT
20 B=PEEK(55)+256*PEEK(56):IFPEEK(B+2)=154THEN40
30 B=B-630:B2=INT(B/256):B1=B-256*B2:POKE55,B1:POKE56,B2:POKEB+2,154:CLR
40 B=PEEK(55)+256*PEEK(56):A=B
50 FORL=5000TO5370STEP10:S=0:PRINT"[SU]LINEA"L
60 FORI=1TO10:READV:IFV>255THEN90
70 S=S+V+1:POKEA,V:A=A+1:NEXTI:READV:IFV<>STHEN90
80 NEXTL:READV:GOTO100
90 PRINT"ERRORE NEI DATA":PRINT"ALLA LINEA "L:END
100 FORL=5390TO5640STEP10:PRINT"[SU]LINEA"L

```

```

110 READX:READY:READV:IFV<>X+YTHEN90
120 X=X+B+1:Y=Y+B:B2=INT(Y/256):B1=Y-
256*B2:POKE X,B1:POKE X+1,B2:NEXTL
130 GOSUB1000:IFPEEK(55)=PEEK(806)AND
PEEK(56)=PEEK(807)THEN160
140 POKEB+324,PEEK(806):POKEB+325,PEE
K(807):POKEB+369,0
150 POKE806,PEEK(55):POKE807,PEEK(56)
160 PRINT"CONVERTITORE INSTALLATO":EN
D
1000 A=B+373
1010 C=32:C$="SPZ":GOSUB4000
1020 C=5:C$="BNC":GOSUB4000
1030 C=17:C$="GIU":GOSUB4000
1040 C=18:C$="RVS ON":GOSUB4000
1050 C=19:C$="HOME":GOSUB4000
1060 C=28:C$="ROSS":GOSUB4000
1070 C=29:C$="DESTR":GOSUB4000
1080 C=30:C$="VRD":GOSUB4000
1090 C=31:C$="BLU":GOSUB4000
1095 C=92:C$="ST.":GOSUB4000
1100 C=96:C$="SHFT *":GOSUB4000
1110 C=123:C$="SHFT +":GOSUB4000
1120 C=124:C$="CMD -":GOSUB4000
1130 C=125:C$="SHFT -":GOSUB4000
1140 C=126:C$="PI":GOSUB4000
1150 C=127:C$="CMD *":GOSUB4000
1160 C=133:C$="F1":GOSUB4000
1170 C=134:C$="F3":GOSUB4000
1180 C=135:C$="F5":GOSUB4000
1190 C=136:C$="F7":GOSUB4000
1200 C=137:C$="F2":GOSUB4000
1210 C=138:C$="F4":GOSUB4000
1220 C=139:C$="F6":GOSUB4000
1230 C=140:C$="F8":GOSUB4000
1240 C=144:C$="NERO":GOSUB4000
1250 C=145:C$="SU":GOSUB4000
1260 C=146:C$="RVS OFF":GOSUB4000
1270 C=147:C$="CLR":GOSUB4000
1280 C=148:C$="INST":GOSUB4000
1290 C=156:C$="PORP":GOSUB4000
1295 C=157:C$="SIN.":GOSUB4000
1300 C=158:C$="GIA":GOSUB4000
1310 C=159:C$="CYN":GOSUB4000
1320 C=160:C$="SHFT SPZ":GOSUB4000
1325 C=168:C$="CMD ST.":GOSUB4000
1330 C=169:C$="SHFT ST.":GOSUB4000
1340 C=186:C$="SHFT @":GOSUB4000
1350 C=20:C$="DEL":GOSUB4000
1998 IFA>B+627THENPRINT"TROPPO LUNGA"
:END
1999 RETURN
4000 POKEA,C:A=A+1:FORZ=1TOLEN(C$):PO
KEA,ASC(MID$(C$,Z,1)):A=A+1:NEXTZ
4010 POKEA,0:A=A+1:POKEA,0:RETURN
5000 DATA72,165,154,201,2,48,4,201,5,
48,910

```

```

5010 DATA4,104,76,67,125,104,141,12,3
,72,718
5020 DATA152,72,138,72,162,3,189,3,0,
168,969
5030 DATA189,113,125,157,3,0,152,157,
113,125,1144
5040 DATA202,16,239,173,12,3,201,255,
208,4,1323
5050 DATA169,126,208,14,201,224,144,4
,233,64,1397
5060 DATA208,6,201,192,144,2,233,96,1
97,5,1294
5070 DATA208,17,238,3,0,120,248,165,4
,24,1037
5080 DATA105,1,133,4,216,88,24,144,85
,172,982
5090 DATA5,0,133,5,165,3,240,53,140,6
,760
5100 DATA0,162,0,189,117,125,240,94,1
97,6,1140
5110 DATA240,9,232,189,117,125,208,25
0,232,208,1820
5120 DATA238,201,32,208,6,165,3,201,2
,144,1210
5130 DATA112,32,28,125,232,189,117,12
5,240,6,1216
5140 DATA32,67,125,24,144,244,169,93,
32,67,1007
5150 DATA125,169,1,133,3,133,4,165,5,
201,949
5160 DATA13,208,11,32,67,125,169,0,13
3,3,771
5170 DATA133,4,133,5,162,3,189,113,12
5,168,1045
5180 DATA189,3,0,157,113,125,152,157,
3,0,909
5190 DATA202,16,239,104,170,104,168,1
04,24,96,1237
5200 DATA240,205,165,6,56,233,161,144
,30,133,1383
5210 DATA6,32,28,125,162,0,189,101,12
5,240,1018
5220 DATA6,32,67,125,232,208,245,174,
6,0,1105
5230 DATA189,70,125,32,67,125,24,144,
163,165,1114
5240 DATA6,201,97,144,27,32,28,125,16
2,0,832
5250 DATA189,107,125,240,6,32,67,125,
232,208,1341
5260 DATA245,165,6,56,233,32,32,67,12
5,24,995
5270 DATA144,221,165,6,32,67,125,206,
3,0,979
5280 DATA208,246,240,172,169,91,32,67
,125,165,1525
5290 DATA4,201,2,48,27,74,74,74,24

```

0,828
 5300 DATA6,24,105,48,32,67,125,165,4,
 41,627
 5310 DATA15,24,105,48,32,67,125,169,3
 2,32,659
 5320 DATA67,125,96,76,42,163,75,73,84
 ,64,875
 5330 DATA71,43,77,92,92,78,81,68,90,8
 3,785
 5340 DATA80,65,69,82,87,72,74,76,89,8
 5,789
 5350 DATA79,64,70,67,88,86,66,67,79,7
 7,753
 5360 DATA68,32,0,83,72,70,84,32,0,1,4
 52
 5370 DATA1,10,46,32,83,80,67,0,5,87,4
 21
 5380 DATA256
 5390 DATA12,323,335
 5400 DATA30,369,399
 5410 DATA37,369,406
 5420 DATA103,373,476

5430 DATA113,373,486
 5440 DATA131,284,415
 5450 DATA135,373,508
 5460 DATA140,323,463
 5470 DATA148,323,471
 5480 DATA163,323,486
 5490 DATA176,369,545
 5500 DATA183,369,552
 5510 DATA211,284,495
 5520 DATA216,357,573
 5530 DATA221,323,544
 5540 DATA230,326,556
 5550 DATA233,323,556
 5560 DATA245,284,529
 5570 DATA250,363,613
 5580 DATA255,323,578
 5590 DATA266,323,589
 5600 DATA274,323,597
 5610 DATA286,323,609
 5620 DATA304,323,627
 5630 DATA314,323,637
 5640 DATA319,323,642

CM 64 CATALOGO-2 PP PP

Un completo programma che Vi permetterà di gestire gli SPRITE, partendo dai valori numerici che normalmente vengono posti dopo le parole chiave DATA all'interno dei programmi. In pratica esso Vi permette di apportare delle modifiche agli SPRITE creati da altri programmatori i posti nel programma nei DATA. Vi permetterà anche di risalire alla forma ed al tipo di un particolare SPRITE, semplicemente ricopiando i valori dalle DATA del programma che si sta esaminando. Un fattore interessante di questo programma è che qualsiasi variazione apporterete ai valori caricati a video ed istantaneamente varierà anche lo SPRITE riportato in alto sulla sinistra. Premendo la lettera H potrete visualizzare le possibili opzioni previste dal programma. Ovviamente tale videata è solo esplicativa, non sarà quindi possibile richiamare le varie opzioni direttamente da essa, ma dovrà essere premuta la BARRA SPAZIATRICE per tornare alla videata di base, quella dove sono visualizzati i valori caricati ed in alto sulla sinistra lo SPRITE come lo avete lasciato prima di premere la lettera H. Le varie opzioni, rappresentate dalla lettera da premere, sono riportate nella riga in alto del video.

SPIEGAZIONE PROGRAMMA

ISTRUZIONI 100-210: Vengono inizializzate le variabili principali. E' interessante soffermarci sull'istruzione 160, ed in particolare sui dati

caricati nella variabile alfanumerica ME\$. Essi corrispondono alle lettere da premere per pervenire alle varie opzioni previste dal menu'. Modificandoli, modificheremo anche le lettere da premere per pervenire alle varie opzioni del menu'. La gestione del tutto viene effettuata nelle istruzioni che vanno dalla 230 alla 280.

ISTRUZIONI 220-280: Come precedentemente detto, queste istruzioni si occupano di gestire il menu' principale, analizzando che le lettere premute sulla tastiera, corrispondano a quelle previste dal programma. Tale controllo e gestione è effettuato, scorrendo la stringa alfanumerica, contenuta nella variabile ME\$, mediante la funzione MID\$ dell'istruzione 240 e 250.

ISTRUZIONI 290-460: Viene gestita l'opzione che si occupa di caricare dalla unità a dischetti i valori della SPRITE in precedenza memorizzata.

ISTRUZIONI 470-620: Viene gestito il menu' secondario, dedicato alla gestione del MULTICOLOR. Viene utilizzata la medesima tecnica utilizzata per il menu' principale, mediante la stringa alfanumerica contenuta nella variabile MM\$ situata nell'istruzione 160.

ISTRUZIONI 630-650: Viene gestito il colore del bordo dello schermo video.

ISTRUZIONI 660-680: Viene gestita l'opzione 0, prevista dal menu' secondario, che gestisce il MULTICOLOR.

ISTRUZIONI 690-710: Viene gestita l'opzione 1, prevista dal menu' secondario, che gestisce il MULTICOLOR.

ISTRUZIONI 720-740: Viene gestita la scelta tra

MULTICOLOR e NORMALE prevista sempre dal menu' secondario.

ISTRUZIONI 750-770: Viene gestita l'espansione orizzontale della SPRITE, attualmente visualizzata.

ISTRUZIONI 780-800: Viene gestita l'espansione verticale della SPRITE, attualmente visualizzata.

ISTRUZIONI 810-1050: Viene gestita l'immissione dei valori numerici, necessari alla creazione della SPRITE e contemporaneamente, viene visualizzata la SPRITE mentre si forma via, via che vengono caricati i valori numerici.

ISTRUZIONI 1060-1190: Viene gestita la lettura della DIRECTORY dell'unita' a dischetti, per dar modo di visualizzare il contenuto del dischetto in quel momento presente in essa.

ISTRUZIONI 1200-1320: Viene gestito il salvataggio della SPRITE, creata sul dischetto presente attualmente nell'unita' a dischetti.

ISTRUZIONI 1330-1350: Vengono gestite le possibili variazioni di colore dei caratteri presenti sul video.

ISTRUZIONI 1360-1370: Viene gestita la fine del programma e la cancellazione di quanto in quel momento e' presente sul video.

ISTRUZIONI 1380-1520: Viene gestito l'aiuto all'utilizzatore del programma, fornendo a video le spiegazioni delle possibili opzioni previste dal menu' principale.

ISTRUZIONI 1530-1770: Viene gestita la modifica dei valori numerici precedentemente caricati e presenti in quel momento sul video; ad ogni modifica d'essi che venga effettuata, verra' contemporaneamente modificata anche la SPRITE presente sul video alla sinistra dei valori numerici.

ISTRUZIONI 1780-1880: Viene gestita la stampa dei valori e della SPRITE, presenti in quel momento sul video, sulla stampante originale del COMMODORE 64.

ISTRUZIONI 1890-1920: Viene gestito il messaggio: "Spazio X continuare", rimanendo in attesa fino a quando esso non venga premuto.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : CATALOGO-2

VAR. : NUMERO ISTRUZIONE

A\$	340	1090	1100	1110	1120	1130
	1140	1150	1160	1250		
B\$	340	350	1120	1130	1250	1260
BC	170	190	640	650		
BI	1820	1830	1840	1850		
BI\$	1820	1830	1840	1860		
BL\$	140	300	350	480	610	820

I	910	980	1070	1170	1210	1260
	1390	1540	1610	1650	1790	
C\$	340	1250				
CL\$	150	210	460	1050	1190	1320
	1520	1740	1770	1880		
COL	170	200	1340	1350		
CU\$	130	910	980	1580	1650	
D\$	340	1250				
FL\$	310	320	330	1220	1230	1240
	1800					
I	180	240	250	270	320	380
	400	430	440	560	570	580
	600	610	840	860	1020	1040
	1230	1290	1300	1560	1580	1590
	1730	1760	1810	1820	1860	
J	420	430	870	1000	1020	1030
	1570	1580	1590	1730	1750	1820
LS	900	910	920	940	980	1640
	1650	1660	1680			
MO	170	670	680			
M1	170	700	710			
ME\$	160	240	250			
MM\$	160	560	570	580		
SD	970	1020	1710	1720	1730	1820
	1830	1840				
SD\$	880	900	930	970	980	1620
	1640	1670	1710			
SDI	170	380	430	440	1020	1300
	1580	1590	1730	1820	1860	
VC	170	190	200	390	650	680
	710	730	740	760	770	790
	800	1070	1190	1350	1370	1390
	1520					
X\$	240	250	570	580	910	930
	940	950	1590	1600	1650	1655
	1670	1680	1690	1900		

100 REM

110 REM

130 CU\$="[13 SIN.]"

140 BL\$="[RVS ON][40 SPZ]"

150 CL\$="[RVS ON][SHFT L][2 SPZ][SHFT M][2 SPZ][SHFT X][2 SPZ][SHFT Y][2 SPZ][SHFT N][2 SPZ][SHFT D][2 SPZ][SHFT S][2 SPZ][SHFT C][2 SPZ][SHFT Q][2 SPZ][SHFT E][2 SPZ][SHFT P][2 SPZ][SHFT H] ELP[3 SPZ]"

160 ME\$="LMXYNDSCQHEP":MM\$="BC01MR":PO KE53272,23

170 DIMSD(63):VC=53248:COL=1:BC=0:M0=2:M1=5

180 FORI=1TO63:POKE703+I,0:NEXT

190 POKEVC+32,9:POKEVC+33,BC:PRINTCHR\$(129)

200 POKE2040,11:POKEVC+39,COL:POKEVC+21,1:POKEVC,70:POKEVC+1,140

```

210 PRINTCHR$(147)CL$:GOSUB390
220 REM
230 GOSUB1900
240 FORI=1TOLEN(ME$):IFX$=MID$(ME$,I,1)
    THEN270
250 IFX$=CHR$(ASC(MID$(ME$,I,1))+128)T
    HEN270
260 NEXT:GOTO230
270 ON I GOSUB300,480,760,790,820,1070
    ,1210,1340,1370,1390,1540,1790
280 GOTO230
290 REM
300 PRINT"[CLR]"BL$"[HOME][RVS ON][SHF
    T N]OME [SHFT F]ILE";
310 INPUTFL$
320 FORI=LEN(FL$)TO1STEP-1:IFMID$(FL$,
    I,1)=" "THENFL$=LEFT$(FL$,I-1):NEXT
330 OPEN15,8,15:OPEN1,8,4,FL$+",S,R"
340 INPUT#15,A$,B$,C$,D$:IFVAL(A$)=0TH
    EN380
350 PRINT"[HOME]"BL$"[HOME][RVS ON]"B$
    " : ";GOSUB1920
360 GOSUB1900
370 GOTO390
380 FORI=1TO63:INPUT#1,SD(I):POKE703+I
    ,SD(I):NEXT
390 POKEVC+21,1:PRINT"[HOME][2 GIU']"T
    AB(23)"[RVS ON][SHFT D]ATA"
400 FORI=1TO63STEP3
410 PRINTTAB(20);
420 FORJ=0TO1
430 PRINTSD(I+J)," ";NEXT
440 PRINTSD(I+2)
450 NEXT
460 PRINT"[HOME]"CL$:CLOSE15:CLOSE1:RE
    TURN
470 REM
480 PRINT"[CLR]"BL$"[HOME][RVS ON][SHF
    T M]ULTICOLOR: [SHFT B][3 SPZ][SHFT C]
    [3 SPZ][0[3 SPZ][1[3 SPZ][SHFT M][3 SPZ]
    [SHFT R]"
490 PRINT"[HOME][2 GIU']"TAB(15)"[SHFT
    B] = [SHFT C]OLORE DEL FONDO"
500 PRINT"[GIU']"TAB(15)"[SHFT C] = [S
    HFT C]OLORE [SHFT S]PRITE"
510 PRINT"[GIU']"TAB(15)"0 = [SHFT M]U
    LTICOLOR #0"
520 PRINT"[GIU']"TAB(15)"1 = [SHFT M]U
    LTICOLOR #1"
530 PRINT"[GIU']"TAB(15)"[SHFT M] = [S
    HFT M]ULTICOLOR SI/NO "
540 PRINT"[GIU']"TAB(15)"[SHFT R] = [S
    HFT R]ITORNO AL MENU'[3 SPZ]"
550 GOSUB1900
560 FORI=1TOLEN(MM$)
570 IFX$=MID$(MM$,I,1)THEN600
580 IFX$=CHR$(ASC(MID$(MM$,I,1))+128)T

```

```

HEN600
590 NEXT:GOTO550
600 ON I GOSUB640,1340,670,700,730
610 IFI=6THENPRINT"[CLR]"BL$:GOSUB390:
    RETURN
620 GOTO550
630 REM
640 BC=BC+1:IFBC=16THENBC=0
650 POKEVC+33,BC:RETURN
660 REM
670 MO=MO+1:IFMO=16THENMO=0
680 POKEVC+37,MO:RETURN
690 REM
700 M1=M1+1:IFM1=16THENM1=0
710 POKEVC+38,M1:RETURN
720 REM
730 IFPEEK(VC+28)=0THENPOKEVC+28,1:RET
    URN
740 POKEVC+28,0:RETURN
750 REM
760 IFPEEK(VC+29)=0THENPOKEVC+29,1:RET
    URN
770 POKEVC+29,0:RETURN
780 REM
790 IFPEEK(VC+23)=0THENPOKEVC+23,1:RET
    URN
800 POKEVC+23,0:RETURN
810 REM
820 PRINT"[HOME]"BL$"[HOME][RVS ON][SH
    FT S]OLO NUMERI PREGO[2 SPZ]"
830 PRINT"[HOME][3 GIU']";
840 FORI=1TO21:PRINTTAB(20)"[16 SPZ]":
    NEXT
850 PRINT"[HOME][2 GIU']"TAB(23)"[RVS
    ON][SHFT D]ATA":PRINTTAB(21);
860 FORI=1TO63STEP3
870 FORJ=0TO2
880 SD$=""
890 PRINT"[COMD @][SIN.]";
900 GOSUB1900:LS=LEN(SD$)
910 IFX$=""THENPRINTLEFT$(CU$,LS)MID$
    (BL$,2,LS+1)LEFT$(CU$,LS+1);GOTO880
920 IFLS=3THENGOTO950
930 IFX$>="0"ANDX$<="9"THENS$=SD$+X$:
    PRINTX$;
940 IFX$=CHR$(13)ANDLS=0THENPRINT"0";
950 IFX$=CHR$(13)GOTO970
960 GOTO890
970 SD=VAL(SD$):IFSD<=255THEN1000
980 LS=LEN(SD$):PRINTLEFT$(CU$,LS)MID$
    (BL$,2,LS+1)LEFT$(CU$,LS+1);
990 GOTO880
1000 IFJ<2THENPRINT" , ";GOTO1020
1010 PRINT" ";
1020 SD(I+J)=SD:POKE703+I+J,SD
1030 NEXTJ:PRINT:PRINTTAB(21);
1040 NEXTI

```

```

1050 PRINT"[CLR]"CL$:GOSUB390:RETURN
1060 REM
1070 PRINT"[CLR]"BL$:PRINTTAB(5);POKE
VC+21,0
1080 OPEN1,8,0,"$0"
1090 GET#1,A$,A$
1100 GET#1,A$,A$
1110 IFA$="THEN1170
1120 GET#1,A$,B$
1130 PRINTASC(A$+CHR$(0))+ASC(B$+CHR$(
0))*256;
1140 GET#1,A$
1150 IFA$="THENPRINT:PRINTTAB(5);GOT
O1100
1160 PRINTA$;GOTO1140
1170 CLOSE1:PRINT"[HOME]"BL$"[HOME]";:
GOSUB1920
1180 GOSUB1900
1190 PRINT"[CLR]"CL$:POKEVC+21,1:GOSUB
390:RETURN
1200 REM
1210 PRINT"[HOME]"BL$"[HOME][RVS ON][S
HFT N]OME [SHFT F]ILE";
1220 INPUTFL$
1230 FORI=LEN(FL$)TO1STEP-1:IFMID$(FL$
,I,1)=" THENFL$=LEFT$(FL$,I-1):NEXT
1240 OPEN15,8,15:OPEN1,8,4,FL$+"$,S,W"
1250 INPUT#15,A$,B$,C$,D$:IFVAL(A$)=0T
HEN1290
1260 PRINT"[HOME]"BL$"[HOME][RVS ON]"B
$": ";GOSUB1920
1270 GOSUB1900
1280 GOTO1320
1290 FORI=1TO63
1300 PRINT#1,SD(I)
1310 NEXT
1320 PRINT"[HOME]"CL$:CLOSE15:CLOSE1:R
ETURN
1330 REM
1340 COL=COL+1:IFCOL=16THENCOL=0
1350 POKEVC+39,COL:RETURN
1360 REM
1370 PRINT"[CLR]";:POKEVC+21,0:POKE532
72,21:END
1380 REM
1390 PRINT"[CLR]"BL$"[HOME][RVS ON][SH
FT H]ELP: ";GOSUB1920:POKEVC+21,0
1400 PRINT"[GIU']][RVS ON][SHFT L][RVS
OFF] = [SHFT C]ARICAMENTO SPRITE[3 SPZ
]"
1410 PRINT"[GIU']][RVS ON][SHFT M][RVS
OFF] = [SHFT M]ODO [SHFT M]ULTICOLOR "
1420 PRINT"[GIU']][RVS ON][SHFT X][RVS
OFF] = [SHFT E]SPANDE IN DIREZ, [SHFT
X][2 SPZ]"
1430 PRINT"[GIU']][RVS ON][SHFT Y][RVS
OFF] = [SHFT E]SPANDE IN DIREZ. [SHFT
Y][2 SPZ]"
1440 PRINT"[GIU']][RVS ON][SHFT N][RVS
OFF] = [SHFT I]MMISSIONE NUOVI DATI"
1450 PRINT"[GIU']][RVS ON][SHFT D][RVS
OFF] = [SHFT M]ENU [SHFT D]ISCO"
1460 PRINT"[GIU']][RVS ON][SHFT S][RVS
OFF] = [SHFT S]ALVATAGGIO SPRITE "
1470 PRINT"[GIU']][RVS ON][SHFT C][RVS
OFF] = [SHFT C]AMBIO COLORI[6 SPZ]"
1480 PRINT"[GIU']][RVS ON][SHFT E][RVS
OFF] = [SHFT E]DITA DATI SPRITE"
1490 PRINT"[GIU']][RVS ON][SHFT P][RVS
OFF] = [SHFT S]TAMPA[6 SPZ]"
1500 PRINT"[GIU']][RVS ON][SHFT Q][RVS
OFF] = [SHFT F][SHFT I][SHFT N][SHFT E
][SHFT SPZ][SHFT L][SHFT A][SHFT V][SH
FT O][SHFT R][SHFT O][SHFT SPZ]"
1510 GOSUB1900
1520 PRINT"[CLR]"CL$:POKEVC+21,1:GOSUB
390:RETURN
1530 REM
1540 PRINT"[HOME]"BL$"[HOME][RVS ON]SP
AZIO PER CONTIN. [SHFT R][SHFT E][SHFT
T][SHFT U][SHFT R][SHFT N][SHFT SPZ]X
SELEZ. "
1550 PRINT"[HOME][3 GIU']"TAB(20);
1560 FORI=1TO63STEP3
1570 FORJ=0TO2
1580 PRINT"[RVS ON]"SD(I+J)LEFT$(CU$,L
EN(STR$(SD(I+J)))+1);
1590 GOSUB1900:IFX$=" THENPRINT"[RVS
OFF]"SD(I+J)"[DESTR]";GOTO1750
1600 IFX$<>CHR$(13)THEN1590
1610 PRINT"[HOME]"BL$"[HOME][RVS ON][S
HFT I]MMETTERE DATI: ";
1620 SD$=""
1630 PRINT"[COMD @[SIN.]";
1640 GOSUB1900:LS=LEN(SD$)
1650 IFX$=" THENPRINTLEFT$(CU$,LS)MID
$(BL$,2,LS+1)LEFT$(CU$,LS+1);
1655 IFX$=" GOTO1620
1660 IFL$=3THENGOTO1680
1670 IFX$="0"ANDX$<="9"THENS$=SD$+X$
:PRINTX$;
1680 IFX$=CHR$(13)ANDLS=0THENPRINT"0";
GOTO1710
1690 IFX$=CHR$(13)THEN1710
1700 GOTO1630
1710 SD=VAL(SD$)
1720 IFS$>255THENPRINT"[4 SIN.][RVS ON
][5 SPZ][4 SIN.]";GOTO1620
1730 SD(I+J)=SD:POKE703+I+J,SD
1740 PRINT"[CLR]"CL$:GOSUB390:RETURN
1750 NEXTJ:PRINT:PRINTTAB(20);
1760 NEXTI
1770 PRINT"[HOME]"CL$:RETURN
1780 REM

```

```

1790 PRINT"[HOME]"BL$"[HOME][RVS ON][S
HFT S][SHFT T][SHFT A][SHFT M][SHFT P]
[SHFT A][2 SHFT SPZ]...":OPEN4,4
1800 PRINT#4,FL$:PRINT#4
1810 FORI=1TO61STEP3
1820 FORJ=0TO2:BI=128:BI$="":SD=SD(I+J
)
1830 IFSD<BITHENBI$=BI$+" ":BI=BI/2:GO
TO1850
1840 BI$=BI$+"[SHFT Q]":SD=SD-BI:BI=BI
/2

```

```

1850 IFBI>=1THEN1830
1860 PRINT#4,BI$;:NEXT:PRINT#4,SD(I)SD
(I+1)SD(I+2)
1870 NEXT
1880 CLOSE4:PRINT"[HOME]"CL$:RETURN
1890 REM
1900 GETX$:IFX$=""THEN1900
1910 RETURN
1920 PRINT"[RVS ON][SHFT S]PAZIO X CON
TINUARE[4 SPZ]":RETURN

```

**CM
64**

FUMO

**PP
PP**

Questo programma e' stato creato per far ragionare le persone sui rischi che procura il fumo. Forse non tutti sanno che a causa del fumo sono aumentati i casi di CANCRO POLMONARE. Il personaggio del programma si chiama Joe ed e' un accanito fumatore come si puo' notare una volta che sia stato lanciato il programma. Ma il troppo fumo ha fatto ammalare Joe... Dalle radiografie si vede come i suoi polmoni si stiano atrofizzando sempre di piu'... Sino a quando Joe muore ed appare la scritta: "NON FUMATE!!!"

SPIEGAZIONE PROGRAMMA

ISTRUZIONE 10: Viene modificato il colore del fondo e del bordo del video e vengono inizializzate le variabili.

ISTRUZIONE 20: Viene stampata a video la scritta "ATTENZIONE FUMARE E' PERICOLOSO". Viene stampata in bianco, su schermo bianco; in questo modo rimane invisibile, sino a quando non passa sopra d'essa la SPRITE che l'evidenzia.

ISTRUZIONI 30-90: Mediante esse, vengono create le SPRITE utilizzate dal programma, leggendo i valori numerici riportati dopo la parola chiave DATA nelle istruzioni che vanno dalla 330 alla 600, mediante la parola chiave READ.

ISTRUZIONI 100-150: Viene gestita la prima parte del programma, quella che si occupa di gestire JOE che sta' SEMPLICEMENTE FUMANDO, facendolo muovere e creando la nuvoletta di fumo.

ISTRUZIONI 160-200: Viene gestita la seconda parte del programma, quella che si occupa di visualizzare i polmoni di JOE ai raggi X, mentre si ammalano di CANCRO ed a visualizzare il messaggio "I SUOI POLMONI SI ROVINANO"

ISTRUZIONI 210-280: Viene gestita la terza parte del programma, quella in cui JOE muore e scompare dietro all'apparecchiatura a raggi X.

ISTRUZIONI 290-310: Viene gestita la quarta ed

ultima parte del programma, quella che visualizza il viso di JOE con la sigaretta in bocca ed il messaggio di avvertimento: "NON FUMATE!!!".

ISTRUZIONI 320-360: Sono qui' contenuti i valori che una volta caricati in memoria, servono a creare la SPRITE della nuvoletta.

ISTRUZIONI 370-410: Sono qui' contenuti i valori che una volta caricati in memoria, servono a creare la SPRITE della testa di JOE.

ISTRUZIONI 420-460: Sono qui' contenuti i valori che una volta caricati in memoria, servono a creare la SPRITE del corpo di JOE.

ISTRUZIONI 470-510: Sono qui' contenuti i valori numerici che servono, una volta caricati in memoria a creare la SPRITE della sigaretta che JOE sta' fumando.

ISTRUZIONI 520-560: In esse vi sono i valori che servono, una volta caricati in memoria a creare la SPRITE dell'apparecchiatura a raggi X.

ISTRUZIONI 570-600: In esse vi sono contenuti i valori, che servono, una volta caricati in memoria a creare la figura di JOE che si dissolve, distrutta dal male che gli hanno procurato le sigarette.

ISTRUZIONI 610-660: In esse vi sono contenuti i valori da caricare in memoria, e fanno parte della routine in linguaggio macchina che si occupa di gestire l'INTERRUPT e rendere cosi' la SPRITE che passa continuamente sopra alla scritta, posizionata nella parte alta del video, completamente indipendente dal programma BASIC.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : FUMO

VAR. : NUMERO ISTRUZIONE

A	:	30	40
C	:	270	280
CC	:	10	90
I	:	170	200 230

```

K : 680
N : 10
O : 50 100 190 250 300 670
RN : 260
SC : 10 90 110
SP : 110 130 210 670
T : 30 40 60 100 110 190
    : 200 220 260
V : 50 60 70 80 190 200
    : 220 240 250 300 670
Z : 130 140

```

```

10 POKE53281,1:POKE53280,1:N=1:SC=1670
:CC=55942:POKE252,25:POKE254,5
20 PRINT"[CLR][4 GIU'][BNC]ATTENZIONE:
FUMARE E' PERICOLOSO[SHFT z]"
30 FORT=12288T012670:READA:POKET,A:NEX
T
40 FORT=49152T049215:READA:POKET,A:NEX
T
50 V=53248:POKEV+27,255:POKEV+29,25:PO
KEV+23,24:0=V+21
60 FORT=0T07:POKEV+T+39,14:POKET+2040,
T+192:NEXT
70 POKEV,70:POKEV+2,52:POKEV+4,52:POKE
V+6,10:POKEV+8,42:POKEV+10,62
80 POKEV+1,130:POKEV+3,162:POKEV+5,183
:POKEV+7,64:POKEV+9,183:POKEV+11,166
90 POKESC,160:POKECC,1
100 PRINT"[15 GIU']ECCO JOE.";:POKE0,6
:FORT=1T02500:NEXT
105 PRINT"[3 SP]ICHE FUMA ."
110 POKESC,32:SP=39:SYS49152:FORT=1T02
500:NEXT
120 GOSUB670
130 Z=Z+1:IFZ=3THENSP=47
140 IFZ=11THEN160
150 GOTO120
160 PRINT"[SU]JOE CONTINUA FUMARE !!!
." :GOSUB670:POKE252,8:POKE254,2
170 FORI=0T06:GOSUB670:NEXT
180 PRINT"[SU][7 SP]] SUOI POLMONI SI
ROVINANO[2 SP]]. . .":POKE252,5:POKE2
54,1
190 POKE0,51:POKEV+1,145:FORT=29T00STE
P-1:POKET+12547,255
200 FORI=1T0T*25:NEXT:POKEV+1,145:NEXT
210 PRINT"[CLR][6 GIU'][SU]E' LA FINE
DI JOE... MUORE !!! . .":SP=18:GOSUB680
:POKE254,2
220 FORT=0T022:POKEV+3,162+T
230 FORI=1T0100:NEXT:NEXT
240 POKEV+1,120:PRINT"[CLR]"
250 POKEV+23,17:POKE0,1:POKE254,255:GO
SUB680
260 FORT=12288T012350:RN=INT(RND(0)*25
6):POKET,PEEK(T)ANDRN

```

```

270 NEXT:C=C+1
280 IFC<6THEN260
290 PRINT,"[8 GIU'][4 SP]NON FUMATE !
!"
300 POKEV+2,100:POKEV+3,115:POKE0,2
310 GOTO310
320 REM
330 DATA 3, 57, 224, 28, 198, 24, 32,
0, 6, 78, 7, 1, 73, 20, 245, 73, 84, 1
33
340 DATA 142, 86, 201, 136, 84, 137, 1
36, 84, 129, 72, 96, 145, 64, 0, 6, 60
350 DATA 24,56,2,39,192,3,192,0,0,6,0,
0,9,0,0,6,0,0,0,0
360 DATA 48,0,0,72,0,0,48,0
370 REM
380 DATA 103,15,224,0,60,248,0,113,255
,56,255,255,252,234,255
390 DATA 252,191,160,32,255,134,112,25
5,200,16,243,195,56,225,192
400 DATA 16,225,128,8,225,128,132,97,1
28,66,124,0,62,112,0,8
410 DATA 232,0,248,8,128,24,4,96,44,2,
24,198,1,7,3,0,129,0
420 REM
430 DATA 103,3,255,128,4,0,32,8,32,16,
16,32,16,31,192,16,16
440 DATA 64,112,16,64,144,16,128,144,2
1,0,144,17,0,80,17,0,80
450 DATA 17,255,160,8,247,160,7,247,19
2,3,239,0,3,207,0,3
460 DATA 207,0,3,207,0,3,199,0,3,252,2
48,3,254,252
470 REM[2 SP]]SIGARETTE
480 DATA 103,0,1,224,0,7,32,0,29,32,0,
21,32,15,245,32,15,255
490 DATA 224,15,255,224,15,255,224,15,
255,224,15,255,224,15,255
500 DATA 224,15,255,224,15,255,224,15,
255,224,15,255,224,15,255
510 DATA 224,15,255,224,15,255,224,15,
255,224,15,255,224,0,0,0
520 REM
530 DATA 255,127,255,254,252,56,127,24
0,16,31,224,16,15,224,16,15
540 DATA 224,56,15,224,56,15,224,56,15
,240,124,31,248,124,63,252
550 DATA 254,127,255,255,255,192,0,3,1
92,0,3,202,0,3,202,114,83
560 DATA 202,87,83,196,101,35,202,87,3
5,202,85,35,202,0,3,160
570 REM
580 DATA 0,0,0,0,0,0,6,0,0,25,0,0,18,0,0
,12,0,0,0,0,0,0,0
590 DATA 0,0,0,0,0,0,0,10,0,0,18,0,0,1
2,0,0,0,0,0,0,0,0
600 DATA 96,0,0,64,0,0,0,0,0,28,0,0,

```

12,0,0,4,0,0
 610 REM
 620 DATA 120,169,13,141,20,3,169,192,1
 41,21,3,88,96,238
 630 DATA 6,208,198,253,165,253,208,7,2
 06,1,208,165,254,133
 640 DATA 253,198,251,165,251,208,26,24

CM 64 ANTI-SUB GG GG

Siete ai comandi di una nave antisommergibili, ed il Vostro compito e' quello di cercare d'impedire ai vari sommergibili nemici d'infiltrarsi nelle Vostre acque territoriali. Per far cio' utilizzando il pulsante di fuoco del joystick connesso alla porta numero due, lanciate la bomba di profondita'. RicordateVi di lanciarla un po' di tempo prima che, il sommergibile passi sotto di Voi, questo a causa del fatto che la bomba impiega un certo periodo di tempo ad innabissarsi. Per spostare la nave verso destra oppure verso sinistra, utilizzate la manopola del joystick. Potete anche utilizzare la tastiera al posto del joystick; in tal caso le spiegazioni sui tasti da usare sono riportate nel programma.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 10-170: Mediante esse viene riservato dello spazio in memoria per le SPRITE, viene cancellato il video ed inizializzate le variabili.
 ISTRUZIONI 180-200: Vengono assegnati i colori alle SPRITE utilizzate nel programma.
 ISTRUZIONI 210-230: Mediante esse, le SPRITE utilizzate nel programma, vengono espanse in senso orizzontale (asse X) ma non in senso verticale (asse Y).
 ISTRUZIONI 240-260: Viene inizializzato il puntatore delle SPRITE.
 ISTRUZIONI 280-310: Vengono letti i valori delle varie SPRITE, dalle istruzioni che vanno dalla 2000 alla 2570, mediante la parola chiave READ.
 ISTRUZIONI 330-410: Viene inizializzato il sintetizzatore interno del COMMODORE64 per la generazione dei suoni utilizzati durante lo svolgimento del gioco.
 ISTRUZIONI 420-560: Vengono gestite le due opzioni previste dal gioco e cioe': TASTIERA o JOYSTICK.
 ISTRUZIONI 570-630: Viene realizzato il cielo raffigurato nel gioco.
 ISTRUZIONI 640-710: Inizializzazione variabili del programma principale.
 ISTRUZIONI 720-780: Gestiscono il movimento

,173,10,208,74,176,9
 650 DATA 238,10,208,206,11,208,76,57,1
 92,206,10,208,238,11
 660 DATA 208,165,252,133,251,76,49,234
 670 POKEV+1,145:POKEO,SP:GOSUB680:POKE
 O,SP-1:GOSUB680:RETURN
 680 FORK=1T0500:NEXT:RETURN

nelle due direzioni del sottomarino che appare sotto alla nave.

ISTRUZIONI 790-870: Viene gestita la parte principale del programma compresa la gestione delle bombe di profondita' che vengono lanciate dalla nave anti sommergibili.

ISTRUZIONI 880-930: Servono a gestire il sottomarino, unitamente alle istruzioni che vanno dalla 720 alla 780 ed a controllare se esso e' stato colpito dalle bombe di profondita'.

ISTRUZIONI 940-1090: Gestiscono il termine del gioco con i vari messaggi di conclusione; servono anche a controllare se chi sta' utilizzando il gioco, abbia il desiderio o meno di continuare.

ISTRUZIONI 1100-2570: In esse sono contenute le varie SPRITE utilizzate nel programma e suddivise nella seguente maniera:

ISTRUZIONI 1990-2070: nave antisommergibili rivolta verso un lato dello schermo.

ISTRUZIONI 2080-2170: nave antisommergibili rivolta verso il lato opposto a quello precedente.

ISTRUZIONI 2180-2270: sommergibile rivolto verso un lato dello schermo.

ISTRUZIONI 2280-2370: sommergibile rivolto verso il lato opposto dello schermo.

ISTRUZIONI 2380-2470: bomba di profondita'.

ISTRUZIONI 2480-2570: esplosione del sottomarino nel caso esso venga colpito dalla bomba di profondita'.

ISTRUZIONI 2580 in poi: Iniziano da questo punto le varie SUBROUTINES utilizzate dalle varie parti del programma principale e sono:

ISTRUZIONI 2980-3060: Gestione della tastiera.

ISTRUZIONI 3070-4080: Gestione del joystick.

ISTRUZIONI 4090-5110: Gestione movimento nave anti sommergibili.

ISTRUZIONI 5120-6090: Gestione movimento sommergibile.

ISTRUZIONI 6100-7070: Gestione sommergibile colpito.

ISTRUZIONI 7080-8100: Istruzioni tasti utilizzabili nello svolgimento del gioco, nel caso si sia scelto di giocare mediante la tastiera.

ISTRUZIONI 8110-9160: Gestione del movimento della bomba di profondita'.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : ANTI-SUB

VAR. : NUMERO ISTRUZIONE

```

A$ : 540 550 1040 8090
CL$ : 110 160 440 590 980 8010
DC : 9010 9020 9050
DD : 690 5010 5020 5030 9010
DE : 7050
DS : 760 770 6010
DX : 840 3010 3030 3040 4010 4040
DY : 4050 5010 5040
FB : 820 3010 3050 4070
FR : 690 820 830 850 860 910
HT : 690 1010 1020 7010
JS : 4030 4040 4050
K : 300 5070 5080 5090 5100 6040
N : 6050 6060 6070 9080 9090 9100
S : 290 300 610 9040 9050 9060
SB : 340 350 360 370 380 390
SL : 400 410 960 1080 7020 9030
T : 600 610 620 630
TI : 690 740 920 7010
V : 550 560 810
XC : 760 770 3020 3030 3040 3050
XS : 4020 4030 4070 9020 9050
Y$ : 870
YC : 140 200 220 230 660 670
    : 680 710 750 830 850 860
    : 910 920 960 5080 5090 5100
    : 6050 6060 6070 6080 7030 7040
    : 7060 9070 9090 9100 9110 9120
    : 9050 9080
    : 670 5040 5050 5060 5070 9020
    : 760 770 920 6010 6020 6030
    : 6040
    : 1060 1070
    : 830 850 9060 9070

```

```

10 REM
20 REM*****
30 REM
100 POKE52,48:POKE56,48:CLR
110 CL$=CHR$(147)
120 POKE53280,6
130 POKE53281,14
140 V=53248
150 PRINTCHR$(144)
160 PRINTCL$:PRINT:PRINT
170 PRINTTAB(18)"ATTENDERE"
180 REM*****
190 REM
200 POKEV+39,11:POKEV+40,0:POKEV+41,2

```

```

210 REM*****
220 POKEV+29,3
230 POKEV+23,0
240 REM*****
250 REM
260 POKE2042,196
270 REM*****
280 REM
290 FORN=0T0383
300 READK:POKE12288+N,K
310 NEXT
320 REM*****
330 REM
340 S=54272
350 POKES,255
360 POKES+1,10
370 POKES+5,72:POKES+6,0
380 POKES+7,25
390 POKES+8,30
400 POKES+12,12:POKES+13,0
410 POKES+24,15
420 REM*****
430 REM
440 PRINTCL$:PRINT:PRINT:PRINT
450 PRINTTAB(14)CHR$(18)"*****"
460 PRINTTAB(14)CHR$(18)"ANTI - SUB"
470 PRINTTAB(14)CHR$(18)"*****"
480 PRINT:PRINT:PRINT:PRINT
490 PRINTTAB(7)"SCEGLIERE I CONTROLLI[
5 SPZ]:"
500 PRINT
510 PRINTTAB(10)CHR$(18)" *** F1 "CHR$(
146)" - TASTIERA"
520 PRINT
530 PRINTTAB(10)CHR$(18)" *** F3 "CHR$(
146)" - JOYSTICK"
540 GETA$:IFA$=""THEN540
550 SL=ASC(A$)-132:IFSL<10RSL>2THEN540
560 IFSL=1THENGOSUB8000
570 REM*****
580 REM
590 PRINTCHR$(5)CL$
600 S$=CHR$(32)
610 FORN=0T04:S$=S$+S$+S$:NEXT
620 S$=MID$(S$,4)
630 PRINTCHR$(18)CHR$(145)S$
640 REM*****
650 REM
660 POKEV+1,78
670 XD=150:POKEV,XD
680 POKEV+16,0
690 SB=20:HT=0:FR=0:DD=-1
700 POKE2040,192
710 POKEV+21,1
720 REM*****
730 REM
740 IFSB=0THEN950

```

```

750 POKEV+3,100+100*RND(1)
760 T=RND(1):IFT<.5THENPOKE2041,194:DS
=-1:XS=300
770 IFT>=.5THENPOKE2041,195:DS=1:XS=5
780 GOSUB6000
790 REM*****
800 REM
810 ONSLGOSUB3000,4000
820 IFFB=0ANDFR=0THENGOSUB9000
830 IFFR=1THENYC=YC+10:POKEV+5,YC
840 IFDX<>0THENGOSUB5000
850 IFFR=1ANDYC>235THENFR=0:POKEV+21,P
EEK(V+21)-4
860 IFFR=1ANDPEEK(V+30)AND2=2THENGOSUB
7000:GOTO730
870 IFTI<40THEN800
880 REM*****
890 REM
900 GOSUB6000
910 IFFR=1ANDPEEK(V+30)AND8=8THENGOSUB
7000:GOTO730
920 IFXS=0ORXS=320THENPOKEV+21,PEEK(V+
21)-2:SB=SB-1:GOTO730
930 GOTO800
940 REM*****
950 REM FINE
960 POKEV+21,0:POKES+4,0:POKES+11,0
970 PRINTCHR$(144)
980 PRINTCL$:PRINT:PRINT
990 PRINTTAB(15)"FINE GIOCO"
1000 PRINT:PRINT
1010 PRINTTAB(16)"DISTRUTTI";HT
1020 IFHT>15THENPRINT:PRINT:PRINTTAB(1
3)CHR$(18)" OTTIMO[6 SPZ]"
1030 PRINT:PRINT
1040 GETA$:IFA$<>" "THEN1040
1050 REM
1060 PRINTTAB(10):INPUT"ANCORA[7 SPZ](
S/N)";Y$
1070 IFASC(Y$)=83THEN580
1080 POKES+24,0
1090 END
1100 REM*****
1110 REM*****
1900 REM
1910 REM[6 SPZ]*****
1920 REM[6 SPZ]*
1930 REM[6 SPZ]* SPRITE
1940 REM[6 SPZ]*[2 SPZ]DATA
1950 REM[6 SPZ]*
1960 REM[6 SPZ]*****
1970 REM
1980 REM*****
1990 REM*****
2000 REM
2010 DATA 0,0,0,0,0,0,0,0,0
2020 DATA 0,0,0,0,0,0,0,128,0
2030 DATA 0,128,0,0,128,0,0,128,0
2040 DATA 1,160,128,1,160,128,1,160,12
8
2050 DATA 1,160,128,19,241,128,19,241,
128
2060 DATA 255,249,152,255,249,152,127,
255,255
2070 DATA 127,255,255,63,255,254,63,25
5,254,0
2080 REM*****
2090 REM*****
2100 REM
2110 DATA 0,0,0,0,0,0,0,0,0
2120 DATA 0,0,0,0,0,0,0,0,1,0
2130 DATA 0,1,0,0,1,0,0,1,0
2140 DATA 1,5,128,1,5,128,1,5,128
2150 DATA 1,5,128,1,143,200,1,143,200
2160 DATA 25,159,255,25,159,255,255,25
5,254
2170 DATA 255,255,254,127,255,252,127,
255,252,0
2180 REM*****
2190 REM*****
2200 REM
2210 DATA 0,0,0,0,0,0,0,0,0
2220 DATA 0,0,0,0,0,0,0,64,0
2230 DATA 0,64,0,0,64,0,0,64,0
2240 DATA 0,112,0,0,112,0,0,112,0
2250 DATA 0,112,0,3,255,0,3,255,0
2260 DATA 127,255,247,127,255,247,255,
255,255
2270 DATA 255,255,255,127,255,247,127,
255,247,0
2280 REM*****
2290 REM*****
2300 REM
2310 DATA 0,0,0,0,0,0,0,0,0
2320 DATA 0,0,0,0,0,0,0,2,0
2330 DATA 0,2,0,0,2,0,0,2,0
2340 DATA 0,14,0,0,14,0,0,14,0
2350 DATA 0,14,0,0,255,192,0,255,192
2360 DATA 239,255,254,239,255,254,255,
255,255
2370 DATA 255,255,255,239,255,254,239,
255,254,0
2380 REM*****
2390 REM*****
2400 REM
2410 DATA 0,0,0,0,0,0,0,0,0
2420 DATA 0,0,0,0,0,0,0,0,0
2430 DATA 0,0,0,0,0,0,0,0,0
2440 DATA 0,0,0,0,0,0,0,0,0
2450 DATA 0,0,0,0,0,0,0,0,0
2460 DATA 0,126,0,0,126,0,0,255,0
2470 DATA 0,255,0,0,126,0,0,126,0,0
2480 REM*****
2490 REM*****

```

```

2500 REM ESPLOSIONE
2510 DATA 240,56,8,240,60,24,48,62,120
2520 DATA 0,0,120,51,18,120,49,146,0
2530 DATA 1,216,192,1,249,131,31,255,7
2540 DATA 15,254,7,7,255,227,97,255,12
2550 DATA 97,191,0,103,63,131,6,89,131
2560 DATA 0,88,192,0,152,96,94,152,33
2570 DATA 194,17,147,224,17,131,224,0,
0,0
2580 REM*****
2590 REM*****
2900 REM
2910 REM[6 SPZ]*****
2920 REM[6 SPZ]*
2930 REM[6 SPZ]* SUBROUTINES
2940 REM[6 SPZ]*
2950 REM[6 SPZ]*****
2960 REM
2970 REM*****
2980 REM*****
3000 REM
3010 DX=0:FB=1
3020 T=PEEK(197)
3030 IFT=23THENDX=16
3040 IFT=12THENDX=-16
3050 IFT=3THENFB=0
3060 RETURN
3070 REM*****
3080 REM*****
4000 REM JOYSTICK
4010 DX=0:DY=0
4020 T=PEEK(56321)
4030 JS=15-(T AND 15)
4040 IFJS>7THENDX=16
4050 IFJS<7ANDJS>3THENDX=-16
4060 REM
4070 FB=T AND 16
4080 RETURN
4090 REM*****
4100 REM*****
5000 REM
5010 IFSGN(DX)<>SGN(DD) THENDD=-DD:DX=0
5020 IFDD=-1THENPOKE2040,192
5030 IFDD=1THENPOKE2040,193
5040 XD=DX+DX
5050 IFXD<40THENXD=40
5060 IFXD>280THENXD=280
5070 K=XD
5080 IFK<256THENPOKEV+16,PEEK(V+16)AND
254
5090 IFK>255THENK=K-255:POKEV+16,PEEK(
V+16)OR1
5100 POKEV,K
5110 RETURN
5120 REM*****
5130 REM*****

```

```

6000 REM
6010 XS=XS+DS*(10*RND(1)+6)
6020 IFXS<0THENXS=0
6030 IFXS>320THENXS=320
6040 K=XS
6050 IFK<256THENPOKEV+16,PEEK(V+16)AND
253
6060 IFK>255THENK=K-255:POKEV+16,PEEK(
V+16)OR2
6070 POKEV+2,K
6080 POKEV+21,PEEK(V+21)OR2
6090 RETURN
6100 REM*****
6110 REM*****
7000 REM
7010 HT=HT+1:SB=SB-1:FR=0
7020 POKES+11,0:POKES+11,129
7030 POKE2041,197:POKEV+40,2
7040 POKEV+21,PEEK(V+21)-4
7050 FORDE=0TO300:NEXT
7060 POKEV+21,1:POKEV+40,0
7070 RETURN
7080 REM*****
7090 REM*****
8000 REM
8010 PRINTCL$:PRINT:PRINT:PRINT:PRINT
8020 PRINTTAB(5)CHR$(18)" *** Z "CHR$(
146)" - MUOVE A SINISTRA[4 SPZ]"
8030 PRINT
8040 PRINTTAB(5)CHR$(18)" *** X "CHR$(
146)" - MUOVE A DESTRA[7 SPZ]"
8050 PRINT
8060 PRINTTAB(5)CHR$(18)" *** F7 "CHR$(
146)" - GETTA LA MINA[5 SPZ]"
8070 PRINT:PRINT:PRINT:PRINT
8080 PRINTTAB(6)CHR$(18)" PREMI UN TAS
TO X CONTINUARE"
8090 GETA$:IFA$=" "THENB090
8100 RETURN
8110 REM*****
8120 REM*****
9000 REM
9010 DC=-DD:FR=1
9020 T=XD-10:IFDC=1THENT=T+40
9030 POKES+4,0:POKES+4,129
9040 FORN=0TO16STEP2
9050 XC=T+DC*N
9060 YC = 70 - 10 * SIN (N/16)
9070 POKEV+5,YC
9080 K=XC
9090 IFK<256THENPOKEV+16,PEEK(V+16)AND
251
9100 IFK>255THENK=K-255:POKEV+16,PEEK(
V+16)OR4
9110 POKEV+4,K
9120 POKEV+21,PEEK(V+21)OR4
9130 NEXT

```

9140 RETURN

9150 REM*****

CM
64CAMPO
MINATOGG
GG

Vi trovate in un esteso campo minato e siete stato incaricato di recuperare le mine disseminate qua e la'. Il Vostro compito non e' per niente semplice a causa del fatto che le mine possono innescarsi ed esplodere in ogni momento. Attenzione anche agli alberi: se ci andate a sbattere perderete. Il gioco e' composto da piu' quadri via, via, sempre piu' complicati, in cui si trovano piu' alberi e piu' mine da disinnescare. Una ultima raccomandazione: Attenzione alle mine che si sono innescate e che si trovano nelle vicinanze di altre mine, perche' l'esplosione di quelle innescate fara' saltare anche quelle vicine. Per giocare utilizzate il joystick connesso alla porta numero uno oppure nel caso vogliate utilizzare la tastiera: i tasti CTRL, -, 1 e 2.

DESCRIZIONE PROGRAMMA

ISTRUZIONI 30-45: Viene modificato il colore del bordo e dello sfondo, portandolo a nero.

ISTRUZIONE 50: Viene richiamata la routine che si occupa della gestione dei vari livelli di cui e' composto il gioco.

ISTRUZIONI 60-265: Vengono inizializzate le varie variabili e viene visualizzata la videata di partenza del gioco.

ISTRUZIONI 270-430: Vengono gestite le varie videate di cui si compone il gioco e la disposizione causale delle mine e degli alberi nel campo.

ISTRUZIONI 440-470: Vengono gestiti i comandi per muoversi.

ISTRUZIONI 480-580: Vengono gestiti i Vostri movimenti nel campo minato.

ISTRUZIONI 590-700: Viene controllato se siete finito sopra una mina innescata o se avete raccolto una mina non ancora attiva, guadagnando cosi' dei punti.

ISTRUZIONI 710-880: Viene gestita l'esplosione della mina che si e' innescata.

ISTRUZIONI 885-940: Viene gestita la raccolta delle mine non attive, togliendole dal campo minato.

ISTRUZIONI 950-1110: Viene gestita la parte finale del gioco e la richiesta per una nuova partita.

ISTRUZIONI 1410-1430 Subroutine richiamata dall'istruzione 50 e che gestisce i vari livelli del gioco.

9160 REM*****

ISTRUZIONI 2000-2080: Subroutine che gestisce il suono dell'esplosione.

LISTA VARIABILI DEL PROGRAMMA

PROGRAMMA : CAMPO MINATO

VAR. | NUMERO ISTRUZIONE

AJ		895	910	920				
B3(		70	360	660				
B4(		70	360	630				
BC(		70	325	416	417	419		
B6		280	870	890				
BN		330	790	810	930	1000		
BP(		70	320	400	410	680	690	
		700	725	740	780	895	1020	
BS(		70	330	610	640	650	660	
		670	680	690	700	730	740	
		780	895	1020				
BT		150	300	350				
BT(		70	350	360	650			
D		150	210	380	410	419	560	
		680	690	700				
DR		430	460	490	500	980		
GR		770	930	960	2050			
HS		180	1045					
J		130	200	210	320	325	330	
		340	350	360	390	400	410	
		415	416	417	419	740	750	
		780	895	961	1020	1081	1083	
K		850	961	970				
N(		120	550	760				
N1		630	725	730	830			
N2		730	750	820	830			
NB		150	290	320	330	340	390	
		600	740	780	870	895	1020	
NL		200	970	980	990	1430		
NN		410	600	610	630	640	650	
		660	670	680	690	700		
NW		150	280	290	300	310		
P(		110	400	416				
PD		720	760	800				
PX		100	380	500	510	520	980	
PY		100	380	500	530	540	980	
Q\$		250	725					
QW		2010	2020	2025	2030	2050	2070	

```

R      : 450 460 470
R$     : 420 1010 1084 1090 1420 1430
S      : 2020
S$     : 240 725
S(     : 100 120 320 380 500 560
        : 890 961 970 1040
SC     : 150 870 880 920 1045
T      : 1050 2060 2070
TG     : 620 630 650 660
TI     : 430 620 720 820 890 940
        : 961 1040
TQ     : 720 820 890 940 961 1040
TX     : 430 620 820 940 1040
X      : 80 90 100 110 120 550
        : 560 570 740 750
X$     : 725 730 770
X(     : 90 725
XM(     : 70 130 500
XP$    : 230 235 730
XR$    : 260 265 770
Y(     : 80 90 725
YM(     : 70 130 500

```

```

30 REM CAMPO DI MINE
45 POKE53280,0:POKE53281,0
50 GOSUB1410
60 REM---VARIABILI---
70 DIMBT(37),B3(37),B4(37),BP(37),BS(37),XM(4),YM(4),BC(25)
80 DEFFNY(X)=INT((X-1024)/40)
90 DEFFNX(X)=(X-40*FNY(X))-1024
100 DEFFNS(X)=1024+PX+40*PY
110 DEFFNP(X)=1307+INT(34*RND(1))+40*INT(15*RND(1))
120 DEFFNN(X)=PEEK(FNS(X))
130 FORJ=1TO4:READXM(J),YM(J):NEXT
140 DATA0,-1,0,1,-1,0,1,0
150 SC=0:BT=1680:NB=4:NW=0:D=54272
160 PRINT"[CLR]";POKE53272,21
170 PRINT"[RVS ON][BNC]MINE****] PUNTI : 0"
180 PRINT"[RVS ON][BNC]*****][DESTR ]P.MASS. :";HS
190 PRINT"[RVS ON][BNC]***CAMPO][5 SPZ ]:";1
200 PRINT"[RVS ON][BNC][8 SPZ][DESTR ]";:IFNL<>1THENFORJ=1TONL-1:PRINT"[SHFT Z]";NEXT
210 FORJ=1024TO1183:IFPEEK(J)=32THENPOKEJ,160:POKEJ+D,1
220 NEXT
230 XP$="[ROSS][SHFT U][SHFT I][GIU'] [4 SIN.][2 SHFT U][2 SHFT I][GIU'] [6 SIN.][3 SHFT U][3 SHFT I][GIU'] [7 SIN.][3 SHFT @][3 SHFT @][GIU'] [7 SIN.][3 SHFT J][3 SHFT K]"

```

```

235 XP$=XP$+"[ROSS][GIU'] [6 SIN.][2 SHFT J][2 SHFT K][GIU'] [4 SIN.][SHFT J][SHFT K]"
240 S$="[HOME][24 GIU']"
250 Q$="[BNC][40 DESTR]"
260 XR$="[BNC][3 SPZ][GIU'] [4 SIN.][5 SPZ][GIU'] [6 SIN.][7 SPZ][GIU'] [7 SIN.][3 SPZ][3 SPZ][GIU'] [7 SIN.][7 SPZ]"
265 XR$=XR$+"[GIU'] [6 SIN.][5 SPZ][GIU'] [4 SIN.][3 SPZ]"
270 REM
280 BG=0:NW=NW+1:IFNW>11THEN310
290 NB=NB+1.5:IFNW=1THEN330
300 IFNW<6THENBT=BT-180
310 PRINT"[HOME][2 GIU'] [RVS ON]";TAB(15);NW
320 POKEFNS(1),32:FORJ=1TONB:POKEBP(J),32:NEXT
325 FORJ=1TO25:POKEBC(J),32:NEXT
330 BN=INT(NB):FORJ=1TONB:BS(J)=1:NEXT
340 FORJ=1TONB
350 BT(J)=(.4+INT(61*RND(1))/100)*BT
360 B3(J)=BT(J)+.5*BT(J):B4(J)=B3(J)+.25*BT(J)
370 NEXT
380 PX=19:PY=15:POKEFNS(1),90:POKEFNS(1)+D,1
390 FORJ=1TONB
400 BP(J)=FNP(1):IFPEEK(BP(J))<>32THEN400
410 POKEBP(J),87:POKEBP(J)+D,8:NEXT:NN=0
415 FORJ=1TO25
416 BC(J)=FNP(1):IFPEEK(BC(J))<>32THEN416
417 IFPEEK(BC(J)+1)=87THEN416
419 POKEBC(J),86:POKEBC(J)+D,5:NEXT
420 GETR$;IFR$<>" "THEN420
430 DR=0:TX=TI
440 REM---COMANDI---
450 R=(15-(PEEK(56321)AND15))*2
460 IFR<>0THENDR=LOG(R)/LOG(2)
470 IFR=0THEN490
480 REM---MOVIMENTI---
490 IFDR=0THEN600
500 POKEFNS(1),32:PX=PX+XM(DR):PY=PY+YM(DR)
510 IFPX<0THENPX=39
520 IFPX>39THENPX=0
530 IFPY<4THENPY=24
540 IFPY>24THENPY=4
550 X=FNN(1)
560 IFX=32THENPOKEFNS(1),90:POKEFNS(1)+D,1:GOTO600
570 IFX=420RX=86THEN960
580 GOTO890

```

```

590 REM---BOMBE---
600 NN=NN+1:IFNN>INT(NB)THENNN=1
610 IFBS(NN)=0THEN600
620 TG=TI-TX
630 IFTG>B4(NN)THENNN=NN:GOTO720
640 IFBS(NN)>2THEN690
650 IFTG>BT(NN)THENBS(NN)=2
660 IFTG>B3(NN)THENBS(NN)=3
670 IF BS(NN)=1 THEN 450
680 IFBS(NN)=2THENPOKEBP(NN),215:POKEB
P(NN)+D,1:GOTO450
690 IFBS(NN)=3THENPOKEBP(NN),87:POKEBP
(NN)+D,1:BS(NN)=4:GOTO450
700 IFBS(NN)=4THENPOKEBP(NN),215:POKEB
P(NN)+D,1:BS(NN)=3:GOTO450
710 REM---ESPLOSIONI---
720 TQ=TI:PD=0
725 X$="[RVS OFF]" + LEFT$(S$,FNY(BP(N1)
)-2)+LEFT$(Q$,FNX(BP(N1))-1)
730 BS(N1)=0:N2=0:PRINTX$;XP$;
740 FORJ=1TONB:X=PEEK(BP(J)):IFBS(J)=0
THEN760
750 IFX<>87ANDX<>215ANDX<>218THENN2=J
760 NEXT:IFFNN(1)<>90ANDFNN(1)<>218THE
NPD=1
770 PRINTX$;XR$;GR=129:GOSUB2000
780 FORJ=1TONB:IFPEEK(BP(J))=32ANDBS(J)
<>0THENPOKEBP(J),87-128*(BS(J)>1)
790 NEXT:BN=BN-1
800 IFPD=1THEN960
810 IFBN=0THEN840
820 IFN2=0THENTX=TX+(TI-TQ):GOTO450
830 N1=N2:GOTO725
840 PRINT"[HOME][2 GIU'] [RVS ON]";TAB(
20);
850 PRINT"[RVS ON]COMPL. ":FORK=1TO100
:NEXT
870 SC=SC-10*(INT(NB)-B6):IFSC<0THENSC
=0
880 PRINT"[12 SIN.]";SC:GOTO280
885 REM
890 B6=B6+1:TQ=TI:POKEFNS(1),218
895 FORJ=1TONB:IFPEEK(BP(J))=218THENAJ
=BS(J):BS(J)=0
900 NEXT
910 IFAJ=4THENAJ=3
920 SC=SC+10*AJ:PRINT"[HOME][RVS ON]";
TAB(16);SC
930 GR=33:GOSUB2000:BN=BN-1:IFBN=0THEN

```

```

840
940 TX=TX+(TI-TQ):GOTO450
950 REM
960 GR=129:GOSUB2000
961 TQ=TI:FORJ=1TO20:POKEFNS(1),42:FOR
K=1TO25:NEXT:POKEFNS(1),170
970 FORK=1TO25:NEXT:NEXT:POKEFNS(1),32
:NL=NL-1
980 POKE1153+NL,160:DR=0:PX=19:PY=15
990 IFNL=0THEN1045
1000 IFBN=0THEN840
1010 GETR$:IFR$<>" "THEN1010
1020 FORJ=1TONB:IFPEEK(BP(J))=32ANDBS(
J)<>0THENPOKEBP(J),87-128*(BS(J)>1)
1030 NEXT
1040 POKEFNS(1),90:TX=TX+(TI-TQ):GOTO4
50
1045 IFSC>HSTHENHS=SC:PRINT"[HOME][GIU
'] [RVS ON]";TAB(19);HS
1050 FORT=1TO1500:NEXT
1051 PRINT"[HOME][BNC][3 GIU'] [RVS ON]
";TAB(20);"FINE[4 SPZ][GIU'] [BNC][9 SI
N. LUNA NUOVA PARTITA[2 SPZ]]?"
1060 PRINT"(S/N)[4 SIN.]";
1080 PRINT"[RVS ON]S/[RVS OFF]N[3 SIN.
.]";
1081 FORJ=1TO99:NEXT
1082 PRINT"[RVS OFF]S/[RVS ON]N[3 SIN.
.]";
1083 FORJ=1TO99:NEXT
1084 GETR$:IFR$="S"THEN1110
1090 IFR$<>"N"THEN1080
1100 PRINT"[CLR][BNC]CIAO[5 SPZ]":END
1110 GOSUB1410:GOTO150
1410 PRINT"[CLR]L[BNC]IVELLO DI DIFFIC
OLTA'(0-3)"
1420 GETR$:IFR$<"0"ORR$>"3"THEN1420
1430 NL=4-VAL(R$):RETURN
1900 END
2000 REM
2010 QW=54272
2020 FORS=QWTOQW+24:POKEQW,0:NEXT
2025 POKEQW+24,47
2030 POKEQW+5,64+7:POKEQW+6,240
2050 POKEQW+4,GR:POKEQW+1,36:POKEQW,85
2060 FORT=1TO250:NEXT
2070 FORT=15TOOSTEP-1:POKEQW+24,INT(T)
:NEXT
2080 RETURN

```

ANNUNCI GRATUITI-ANNUNCI GRATUITI

Tutti i lettori che desiderano comunicare con il mercato, possono approfittare della rubrica di piccoli annunci gratuiti, compilando e spedendo a: CO.STA EDIZIONI - VIA MAZZINI 12B - 50100 PISA il sottostante tagliando.

___ VENDO	___ COMPRO	___ SCAMBIO	___ INFORMO
___ HARDWARE	___ SOFTWARE	___ ALTRE	

NOME _____			
VIA _____		CITTA' _____	
C.A.P. _____	TELEFONO _____	ORARIO _____	

MERCATINO - MERCATINO - MERCATINO - MERCATINO

Scambio programmi con tutta l'Italia. Scrivere o telefonare a: UMBERTO BENELLI - Viale XX Settembre 180 54031 - CARRARA-AVENZA (MASSA).
Tel. 0585/ 57145 - Telefonare dalle 13 alle 23.

Scambio - Acquisto - Vendo programmi per il COMMODORE 64, il PC IBM e compatibili, il COMMODORE 16 ed il PLUS/4 preferibilmente su dischetto. Scrivere a: Negri Giusy - Via Adone del Cima 39E/12 - 55048 - Torre del Lago Puccini - (LUCCA).

Scambio programmi COMMODORE. Scrivere o telefonare a: LEONARDO DE PALMA - Via Francesco Marinaccio, 7 - 71100 FOGGIA - Tel. 0881/ 79700 - Telefonare nel pomeriggio.

SCAMBIO PROGRAMMI COMMODORE. Scrivere o telefonare a: ROBERTO DSELLADORE - Via Passo San Boldo, 35/2 30030 - FAVARO VENETO (VENEZIA).
Tel. 041/ 631106 - Telefonare dalle ore 12 alle ore 13.30 e dalle ore 17 alle ore 18.15.

Scambio programmi COMMODORE. Scrivere o telefonare a: FRANCESCO DE COLLE - P.le Capo Linaro 11/23 00053 CIVITAVECCHIA (ROMA)
Tel. 0766/ 34171 - Telefonare ore pasti.

Scambio programmi. Scrivere o telefonare a: GIANNI GAGGINO - Via Graglia 18 10136 - TORINO
Tel. 011/ 352830 - Telefonare dalle ore 12.30 alle ore 14.30 Oppure dalle ore 19.30 alle ore 21.30.

Scambio programmi. Scrivere o telefonare a: LUCA LORENZINI - Via Lumumba, 11 41011 - CAMPOGALLIANO (MO)
Tel. 059/ 525861 - Telefonare dalle ore 14 alle ore 15.30 e dalle ore 18 alle ore 21.

Scambio programmi su computer della COMMODORE. Scrivere o telefonare a: MASSIMILIANO GRENANIN - Via XXV Aprile, 4 10036 - SETTIMO TORINESE (TO)
Tel. 011/ 8010392 - Telefonare dalle ore 18 alle ore 20.



PERSONAL COMPUTER **CLUB**

GIOCHI-CONCORSO

PROGRAMMI E TRUCCHI

PROVE HARDWARE E SOFTWARE

PROGETTI SOFTWARE

BIBLIOTECA PERSONALE

e le rubriche

TUTTOCOMMODORE

TUTTOSINCLAIR

TUTTOMSX

**OGNI MESE
IN EDICOLA
LIRE 3.000**